IMPLEMENTATION OF PROBLEM ENVIRONMENTS AND SOFTWARE SUPPORT TOOLS FOR A MANAGEMENT INFORMATION SYSTEM COURSE FOR STUDENTS IN THE COLLEGE OF BUSINESS ADMINISTRATION AT KANSAS STATE UNIVERSITY

BY

CONSTANZA CASTRO

B.S., University of Oregon, 1975
MBA, Kansas State University, 1976

A MASTER'S THESIS

Submitted in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

approved by

[illegible]

Major Professor

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Chapter I will describe the curriculum needs that will be satisfied by the MIS undergraduate course as well as the reasons for inclusion of such a course in the curriculum of the College of Business at Kansas State University. It will also define an MIS and what it should accomplish for an organization.

The College of Business Administration at Kansas State University has started this semester (Spring 1984) to offer a Management Information Systems course (MIS) to satisfy the demands of an updated curriculum for the Bachelors of Science in Business Administration. The teaching of this course during the first year will have to be adapted to the present needs of those potential candidates some of which can be stated as follows:

1. Need to obtain more basic knowledge about Management Information Systems (MIS).
2. Need to interact with actual systems for practical applications of Management Information Systems.
3. Need to become familiar with potential problems and opportunities involved in the actual implementation of an MIS.
4. Need to be able to demonstrate familiarity and knowledge to potential employers who have repeatedly requested the inclusion of an MIS course in the curriculum.

Evaluation of the degree of achievement of the learning objectives during the first semester of the course will affect the course content in future semesters.

1.1 Course History

The College of Business Administration has been planning two Management Information Systems classes geared specifically to the business students:

> MANGT 466 - Management Information Systems. A comprehensive view of the organization's requirements and the role of computer information systems in gathering and producing information. Concepts of data resource management, assessing developments in information technology, and information system's impact on organizations. Problems and techniques concerning the development and installation of responsive systems with special attention to manager's use of system's outputs. Case studies and selected applications.
> Pr.: CMPSC 202, FINAN 450, MANGT 420, and MKTG 400.
>
> MANGT 866 - Advanced Management Information Systems. An indepth, analytical treatment of organizing, producing, and using information in complex organizations. Examination of information-management tools and concepts including technological developments, data processing, information system's impact on organizations, and system output implementation. Problems and techniques concerning the development and installation of responsive systems's output.
> Pr.: MANGT 466 or CMPSC 202, FINAN 450, MANGT 420, and MKTG 400. [1]

These two classes form a sequence, with students getting basic knowledge in the undergraduate course and advancing to more sophisticated concepts and applications at the graduate level.

1.2. Need for an MIS Class at the Undergraduate Level

In the modern business world the need for managers with a certain level of "computer literacy" -- knowing the basics of working with a computer system and ability to interact with Operations Research and Systems staff--has become more and more acute due to several factors such as:

1. Increasing complexity of the management task.

a) The size of organizations has grown tremendously during the past decade. For example, the number of employees in the nation's 500 largest industrial firms increased from 11.3 million in 1965 to 16.2 million in 1979, and assets increased from $242 to $1,035 billion. [2]

b) Increasing complexity of technology employed within the organization (i.e. factory robots, automated merchandise storage and movement, etc.) This requires the effort to keep pace with technology to be continuous.

c) Shrinking time frame for decisions. This increases the need for the managers to act quickly in response to pressures from customers, competition, and stockholders.

d) Social pressure. This adds another dimension to the task of business decision making. Decisions must be based on economic factors, but the manager must have means for assessing social costs and payoffs that must be weighed in the decision.

2. Availability of Decision Making Tools. During the 1950s, efforts to solve business problems with advanced mathematics were called Operations Research (OR). These efforts were usually designed to prevent or solve manufacturing problems. During the 1960s, the term management science became popular, as quantitative methods were applied on a broader scale--in finance and marketing, for example. The increasing popularity of the computer in the late sixties and seventies led to attempts to harness its power for the mathematical computations in Quantitative Management Techniques. Terms such as Management Information Systems (MIS) and Decision Support Systems (DSS) represent currently popular means of assisting the manager with computer-produced information. MIS refers to the overall application of the computer in a firm, with

the emphasis on supporting management's information needs. DSS refers to a subset of the MIS, intended to provide information for the solution of Management Decision Problems relating to specific situations. [3]

1.3 MIS Concept Definition

The student will be expected to become familiar with both the concepts and the tools involved in a successful implementation of a computerized Management Information System (MIS).

Even though some firms still have information systems based on manual record keeping, (using file cabinets to store records, card systems for inventory, manual payroll calculations, etc.), the present availability and wide use of the computer makes it desirable for the student to be able to comprehend and handle the more sophisticated systems. The definition of an MIS be used for the class is as follows:

> An organized method of providing past, present and projection information relating to internal operations and external intelligence. It supports the planning, control, and operational functions of an organization by furnishing uniform information in the proper time frame to assist the decision making process. [3]

The depth of this description can be better appreciated when each component part is considered:

a) Organized method: The MIS is organized. Since the MIS is a system, the parts should work together in an organized manner to achieve efficient performance.

b) Past, present and projection information: Information is provided to the manager to make it possible to appraise where the firm has been where it is now and where it is going. Before computers were available most systems used by the managers were designed to provide only past information. Those systems, using punched card machines, key-driven machines, or manual processes, generated historical reports for the manager. The manager used these reports as a basis for deciding what should be done in the future. The systems were so slow that the manager seldom had a good idea of what was happening presently. By the time present performance was reported, it was past history. An important characteristic, then, of the modern MIS is the ability to report information about the present and the future--information that was generally unavailable before the computer era.

c) Internal and external information: Information is provided about what is happening both inside and outside the firm. Compared to previous systems that provided mainly internal information, the MIS places great value on external, or environmental, information. This environmental information is especially important to top level managers. The president of Ford Motor Company, for example, pays more attention to the effect of Japanese automobile imports than to the many internal matters left to competent administrators.

d) Planning, control, and operation functions: The MIS should help the manager plan what to do, execute plans, and control the firm's activity to assure that the plans are carried out. No part of the manager's activity should be left unsupported by the MIS. The MIS is a broad, comprehensive system, then, in its support of the manager.

e) Uniform information: The MIS should be ongoing, providing information on a continuing basis. This factor separates MIS from other business information gathering activities. such as marketing research or financial research, that are directed at solving a particular problem. Once the problem is solved, the data gathering activity is terminated. An example of uniform information is a monthly report prepared for the sales manager showing the return on investment (ROI) for each of the sales offices.

f) Proper time frame: The information provided by the MIS must be quickly available. This requirement of responsiveness is especially critical for information. Often the conceptual information system must respond immediately to the needs of the physical system--perhaps within seconds. As the size of the firm increases, the need for responsiveness demands a computer--and often a larger and more expensive computer as the need for a fast response intensifies.

g) Assists in decision making: the MIS is designed to help the manager make decisions. The intent is not for the computer to make decisions for the manager, but for the computer to provide information support to the manager while he or she is making decisions to solve problems. This is the decision support system concept. Although managers do not spend all their time making decisions and solving problems, these activities are the focus of the MIS and the DSS. [3]

1.4 Project Definition

The main aim of this thesis work will be the actual implementation of an MIS course to satisfy the student and curriculum needs as defined in this introduction.

Chapter II will discuss the historical background for this thesis, the relevant Student Behavioral Learning Objectives (SBLO), target audience characteristics, and the sequence of assignments designed to achieve these SBLO.

Chapter III and IV will cover the actual software implementation and the detailed analysis of the changes made to the available programs.

Chapter V deals with the evaluation of the implementation and Chapter VI contains the conclusion and recommendations for future work.

CHAPTER II

PROJECT BACKGROUND

In 1982 Neil Strunk developed in his masters project titled "Support Tools for an Undergraduate Level MIS Course" [4] a plan to give business students an adequate preparation in MIS. He covered several areas important to the introduction of a new MIS course for business students. The student Behavioral Objectives (SBLO) and some topics were outlined, potential textbooks and ideal hardware and software were reviewed. The SBLO were used as a foundation for possible problem environments.

In 1983, Robert A. Birchard in another masters project entitled "Design and Implementation of Problem Environments and Software Support tools for a Management Information Systems Course" [5] built on Strunk's research as it could be applied to an MIS course geared to students in the Information Science (IS) curriculum in the Department of Computer Science at Kansas State University. Birchard's main areas of concern were:

1. Specific problem environments to be used.
2. Software requirements for implementing these problem environments.
3. Implementation of the software support tools.

The present project will build upon the work done by Birchard to develop the actual classroom implementation of the SBLO and software support tools in an MIS course designed for students in the undergraduate curriculum in the College of Business Administration at Kansas State University.

2.1 SBLO Definitions

Student Behavioral Learning Objectives (SBLO) are:

1. Statements of what the student is to learn in the MIS course.
2. Standards by which the progress of the student can be measured.[4]

2.2 Audience

The SBLO have to be adapted to the present characteristics of the audience. The students are in their junior or senior year of the business curriculum; they have had several business courses which provide them with a good understanding of the basic business functions. However, their required background in computer applications is limited to the fundamental of computer programming course (CMPSC 202). They have had no previous database experience. Therefore, they need additional guidance in the use of software tools in an interactive environment.

These audience characteristics differ from the ones specified for the IS majors, who were the target of Birchard's work.

2.3 SBLO Selection

The only available hardware for an MIS course in the College of Business Administration is the National 6/30, the mainframe computer available for teaching and research on a campus-wide basis at Kansas State University. This puts a further restriction on the selection of a subset of the SBLO suggested by Strunk [4] (Data Base Management System, Decision Support Systems, Office Automation, Transaction Processing). These SBLO can be found in appendix D. Birchard had worked on possible implementations for the Data Base Management Systems, Decision Support Systems and portions of the Office Automation SBLO. Due to the hardware limitations, the actual implementation of the Office Automation SBLO will not be conducted (Birchard planned to use the Interdata 8/32, the computer available through the Computer Science Department).

The selected SBLO are as follows:

2.3.1 Decision Support Systems

Decision Support Systems (DSS) are interactive computer systems. They are designed as an aid to managers in solving problems that involve judgment (i.e., unstructured and structured problems).

The SBLO for DSS are as follows:

- use an available application program to answer a structured problem.
- use one or more available DSS tools and combine the pieces to make a decision about a predefined problem.
- design a semi-structured problem environment within a predefined scope [5].

2.3.2 Database Management Systems

Objectives defined in the area of Database Management Systems will make use of one database and one database management system. The database will already be available and the SBLO are:

- use a query language to retrieve records.
- use a query language to retrieve information from different types of records
- add records to the database.
- delete records from a database.

- change field values in records in the database. [5].

The database will be provided to the students using IDMS (Integrated Data Management System, Cullinet Corporation) available through the National 6/30. Correct implementation of the above mentioned SBLO constitutes completion of the task.

Furthermore, the students will be required to explore a third area, namely:

2.3.3 Text Editing

The SBLO for this particular task will involve:

- using SCRIPT (text editing tool available through the National 6/30) to edit and format a predefined document.
- Route the formatted document through the system.
- Produce output at one of the printing stations according to specifications.

Completion of these steps will accomplish the SBLO for the task.

(A summary table showing the selected SBLO can be found in Appendix D.)

2.4 Description of Software

2.4.1 Integrated Database Management System (IDMS): a product of Cullinet Corporation, state of the art, network oriented database management system. Version 5.7, currently available at Kansas State University, can be accessed through CMS; by using a front-end program that acts as a query language interface, the student can gain the experience of working with an interactive DBMS.

2.4.2 EMPIRE: a product of Data Research Associates. It is a financial planning, modeling and decision support tool. It can be accessed interactively through CMS; it allows not only data manipulation but the use of more sophisticated analysis methods to obtain further insights into the relationships found in financial analysis: simulation, linear regression, target analysis, graphics and report generation. The language used for model building is a form of simplified Fortran. The other commands used to conduct the analysis are simple and English-like.

2.4.3 SCRIPT: one of IBM word processing tools available through the CMS editing environments. It will be used through the Xedit environment, using some of the most common formatting commands. Script as a word processor is not as "user friendly" as some of the word processing software found at the microcomputer level, but it will give the students an opportunity to obtain hands-on-experience with one of the tools they will have to manipulate in the future.

2.5 SBLO Implementation

The following is a description of the implementation of the SBLO through the utilization of the available software tools.

2.5.1 SCRIPT

This is the first software tool that the students will be required to use. There are two main reasons for this choice:

1. The students will be able to gain familiarity with the CMS system.
2. The student will be able to use it for future work during

the course as it will be required that they type their case analyses using SCRIPT.

The assignment will consist of the following:

1. Log on to the National 6/30 through CMS.
2. Access the XEDIT and SCRIPT environments.
3. Type a document of at least two pages in length using pre-specified SCRIPT commands.
4. Produce a scripted version of the document.
5. Print both the original SCRIPT file and the scripted document at the main computer laboratory in Cardwell Hall.
6. Retrieve output and hand in assignment.

Very specific instructions have been developed for the students. A copy of the documentation can be found in Appendix A.

In addition to fulfilling the SBLO for text editing, this assignment will have an added benefit: the majority of the students in the present course have taken their only computer programming course using batch processing and keypunched cards. This assignment and the case analyses typing will definitely help overcome some of their unfamiliarity with interactive systems, enabling them to approach the other assignments with more confidence.

2.5.2 Decision Support Systems

The EMPIRE assignment will consist of the following:

1. Invoke EMPIRE system through CMS.
2. Execute three models:
   a. Finansum - This model creates a ten-year financial summary for the Justin Allen Corporation. All the necessary model, control, data, and report files will be provided.

b. Model 2 - This model creates a forecast of potential earnings for the coming year for product No. 4792, Chem46; the student will get the necessary files to produce the final report. The student is then required to perform Monte Carlo simulation, target analysis, and impact analysis, to find what would be the best way to improve the earnings per share performance for the product. Graphical representation of the forecast part of the model will also be produced.

c. Model 5 - The same type of financial forecast done for Model 2 but for product number 4000, Newlub. No additional simulation or analysis is required.

3. Print result of EMPIRE session at terminal in Cardwell and hand in for completion of assignment.

The students are also required to consult the EMPIRE manuals to ensure their understanding of the different parts of the models they have worked with.

A very detailed set of instructions have been prepared for this assignment. It can be found in Appendix B.

After this assignment is complete, the students are then required to perform a simple breakeven analysis on their own, using some of the commands that they have learned from the first assignment. This assignment, though simple, is designed to give the student further confidence in his/her ability to work with the models found in EMPIRE without needing an extraordinary amount of direction from the instructor. A sample of the assignment can be found in Appendix B.

The output for the EMPIRE models 2 and 5 will be used as input for the DBMS assignment.

2.5.3 Database Management Systems

The IDMS system will be used by implementing the database and the user front end designed by Robert Birchard, [5] with some modifications discussed later on in this paper. A detailed description of the records in the database can be found in Appendix C.

The assignment description is as follows:

1. Access IDMS through CMS.
2. Retrieve the market estimate (MKTEST) and MARKET records for product Chem46.
3. Change values in the records according to output of EMPIRE model 2.
4. Create and add new records for Newlub PRODUCT, Newlub Midwest MARKET, Midwest 1984 MKTEST, and market history, (MKTHIST), using the output from EMPIRE model 5 for the input figures.
5. Delete two records from the database.
6. Produce a report of the updating session.
7. Print report at Cardwell terminal.
8. Hand in report for completion of assignment.

The assignments will be given in the same sequence as they were described above:

1. SCRIPT assignment
2. EMPIRE assignment
3. DBMS assignment

The experience gained by typing the case analysis using SCRIPT is additional to the formal assignments delineated above.

2.6 Case Analyses

Three case analyses will be required during the semester. Specific instructions as to the format and content of the finished analysis will be given. The students will work in teams to do the actual analysis work. These teams will be composed of a maximum of three students each, to provide the best chance for participation in discussion and cooperation in the production of the final document. There will be in-class discussion of the several solutions presented by the different groups.

The cases cover 3 main areas:

1. Problems found in updating of a data processing operation into an MIS.
2. Dealing with problems caused by lack of proper planning and implementation of a computerized MIS.
3. Using systems flow documentation to represent an existing system.

2.7 Lectures

The topics will give comprehensive coverage of the following areas:

1. MIS and DSS components.
2. General systems and management concepts.
3. Computerized systems components.
4. Planning and Implementation of an MIS.
5. Use of the MIS at different levels of management.

(A class syllabus and a summary table of the implemented SBLO can be found in Appendix D.)

2.8 Guest Lecturers

Two guest lecturers will be invited during the semester to give students a better perspective of the issues discussed in class through the experience of practitioners in the areas of computer systems and database management, such as computer services managers and database administration.

2.9 Further Software Analysis

The basic definitions and description of background work covered in this chapter will allow further discussion of the actual implementation of the software tools. Chapter III will cover this area and will also delineate the scope of the implemented SBLO and the chosen sequencing of the assignments designed to achieve these SBLO.

CHAPTER III

ANALYSIS OF SOFTWARE IMPLEMENTATION

The actual classroom implementation of the several software tools required further refinement of the work conducted by Birchard [5] in his project. The following is a description of the changes made to his suggested implementation and the rationale behind those changes.

3.1 SBLO

As mentioned in the software description section of this project, the hardware limitations present at this moment in the College of Business Administration make it impossible to implement the transaction processing and office automation SBLO. Furthermore, the SBLO for the DBMS has been changed as follows:

3.1.1 Scope of SBLO

Birchard [5] specified in his SBLO that the students would be required to add, delete, and change "a record" from the database. These SBLO have been changed to read "records". This change was made to give students increased exposure to the use of a query language.

Another change is the elimination of that portion of the SBLO which related to the manipulation of subschema structure. Birchard [5] did not actually implement this part of the SBLO which would consist of providing the students with the Data Manipulation Language (DML) statements necessary to add to a new set and record types to the database. The students in the College of Business do not possess the background in computer programming to be able to handle this type of assignment, even though it would be perfectly appropriate for the IS majors.

3.2 Assignment Sequence

Due to the enormous amount of storage space required by the files necessary for the two software packages used (EMPIRE and IDMS) (a description of the storage space requirements can be found in Appendix E), the best way found to handle the assignments was to alter the sequence suggested by Birchard which was as follows:

```
DBMS problem 1 (retrieve a record from database)
     DDS problem 1 (Finansum summary).

DBMS problem 2 (retrieve two records from database)
     =====>DSS problem 2 (Model 2 financial projection).

DSS problem 3 (model 2 simulation results)
     =====>DBMS problem 3 (change a record in database)

     (where =====> indicate flow of data)
```

The new sequence is as specified in the SBLO implementation section:

```
EMPIRE assignment (includes implementation of Finansum,
     model 2, model 2 simulation, model 5, and breakeven
     analysis.

     ============>

DBMS assignment (use data from EMPIRE models 2, simulation,
     and model 5 to fulfill SBLO for DBMS (retrieve, change,
     add, and delete several records).

     (where ============> indicates flow of data.)
```

This sequencing of the assignments will have the added advantage of making it possible for the students to concentrate on learning and working with the EMPIRE system before being introduced to IDMS. This is a significant aspect when the concern for audience ability to handle interactive systems is a major factor.

The students will have access to these files through a "library" established through the master class account. All the students in the class will have access to the necessary files simultaneously when they sign on their respective accounts, and their filemanager space will not be crowded with excess information, which would occur if the information were placed in each one of their individual filemanager spaces.

Each student will have 100 blocks of filemanager space available throughout the semester and enough funds to work with the software implementations required.

3.3 Software Tools Refinement

Having delineated the basic changes needed to adapt the previous work done by Strunk [4] and Birchard [5] to make the actual implementation possible. Chapter IV will discuss the refinement of the software tools and the documentation and instructions that will be made available to the students for the fulfillment of the SBLO for the class.

CHAPTER IV

ANALYSIS OF TOOL DEVELOPMENT

The tools used for the MIS course during spring semester, 1984, were based on the implementations initially developed by Robert A. Birchard [5]. His documentation for the changes necessary to the CMS environment in order to work with EMPIRE and IDMS proved to be very useful.

4.1 EMPIRE Software system

Birchard selected those models to implement with the EMPIRE software (Finansum and Model 2) which were also used by Applied Data Research (publishers of the EMPIRE software) to explain the different features of this package. This choice seemed to be appropriate for the actual classroom implementation: students who have not had previous experience working with that type of software can very easily consult the manuals on the specific commands used in the implemented models. Two other models (Model 5 and Model 1) have been designed to augment the documentation available on the corporation being modeled and provide further sources of information to be used for the IDMS assignment.

4.1.1 Documentation availability

EMPIRE manuals at the KSU campus can be found at the following locations: Cardwell Hall (room 23) and Calvin Hall (room 9). The manuals, unfortunately, are not maintained up-to-date by the Computer Services office due to the low usage rate on a campus-wide basis. This requires that additional copies of the respective up-to-date versions of the manual be placed at other locations on campus (i.e. reserve desk at Farrell library).

In addition to this documentation available to students, their limited experience with computers makes detailed verbal and written instructions necessary.

<u>4.1.2 Verbal Instructions</u>

The following instructions will be given in class the day the assignment is handed to students:

1. This is a set of financial analyses for the Justin Allen Corporation.

2. The first model (Finansum) is a financial summary for ten years of operation of the corporation. The model includes a control file that will cause all the manipulations necessary to be done automatically. The only concern of the student is to key the commands in according to the instructions (found in appendix B of this paper). This financial summary will provide the student with background information on the past performance of the Justin Allen Corporation.

3. The second model (Model 2) provides a forecast analysis for product number 4792, Chem46, for the Upstate market during 1984. This model also contains a control file that will automatically execute the basic commands necessary for model manipulation up to producing the actual forecast and typing the report. Since this is a forecast for a new product, the company is interested in finding out which would be the best way to obtain a potential Earnings per Share (EPS) figure of at least \$2.80. The student will execute target analysis and Monte Carlo simulation to reach a conclusion for the best way to achieve this goal. The result of the analysis will be used to update the company's database which at this point contains only the information from a previous forecast.

4. The third model (Model 5) is another forecast for product number 4000, Newlub. This is a new product to be introduced in a new market and the figures obtained from this model will be saved into the company's database for later refinement.

5. The fourth model (Model 1) is a breakeven analysis for analyzing different capital intensive production techniques for another potential product for Justin Allen Corporation. The output from this model will help the company identify the best choice at different levels of output.

4.1.3 Written instructions

The manual prepared for the EMPIRE assignment can be found in Appendix B. Again, due to limited experience the students have with computers, it is extremely detailed in describing the input necessary on the part of the student to obtain the required output. It includes instructions (based on Birchard's specifications [5]) to set up the CMS machine to access the EMPIRE system . This manual differs from Birchard's since the sequence of assignments and the output required are different from his specifications and the level of detail is much greater due to the different target audience.

4.2 IDMS

The database schema and subschema created by Birchard [5] for the Justin Allen corporation were implemented with minor changes. A diagram of the structure of the schema can be found in Appendix C.

The user front-end, however, had to be changed substantially in order to accommodate the difference in the implementation of the SBLO for this assignment.

4.2.1 Description of Changes to front-end program

In addition to the unavoidable problems found when trying to transcribe programs from typed text (Birchard's typed source code did not perform as described in many instances), major modifications to several of his procedures were necessary in order for the program to work as required by the new SBLO.

The following is a detailed description of the changes made to the program to perform the tasks for the required SBLO.

4.2.1.1 Schema updating

After transcription of the program found in Birchard's project, some inconsistencies between the schema and the subschema definitions were found. His written definition of record types and what was actually found in the schema conflicted (i.e. the MKTHIST record was described as having the product id number as one of the fields when it did not have it in the schema. Since this is not necessary for the database implementation, the record was left as found in the schema. This was also the case with the MKTEST record, where the product id number was not included as specified in his record description).

When front-end implementation and development was started, it was found that the schema description lacked one of the fields for the MKTEST record which was necessary for the front-end, namely the MKTSHR field. The schema was recompiled to include it.

4.2.1.2 Front-end Development

The following are the procedures in the front-end program that required changes, with a description of those changes.

1. Main Variable Declarations

Two new global variables were declared (DO_CMD and CHG_OK). These variables were used as flags to control program performance in special cases when user input did not match expectations of the front-end program or IDMS. Without these flags, errors in input from the terminal would cause session termination due to record specification conflicts.

Record lengths for some of the buffer variables were changed to make them agree with the schema description.

2. DBCHG_VAL Procedure.

This procedure changes the field values in a particular record occurrence. The following changes were necessary to facilitate proper processing.

CHG_OK flag was set to "true" at the beginning of the procedure. This change was needed since this procedure is called to change the values in the database records: without setting this flag when a user typed the wrong value, he/she could not correct the error and continue with processing of that particular record since the record changes would not be sent to the IDMS processor by the procedure DBCHANGE.

3. DBCHANGE Procedure.

This is the procedure invoked to initiate changes to the fields in the record occurrences.

A statement was added to insure proper handling of situations in which the user selected the DBCHANGE command by mistake. If the user decides not to change a record, nothing is done and control is returned to the main program. Before, control passed to IDMS and an error situation occurred.

4. DBREPORT Procedure.

This procedure is invoked by the user to store a "picture" of the contents of the entire database at any given moment into the output file created for this purpose (dbout listing).

This report generation procedure was able to handle only a single EST and HIST set with multiple MKTEST and MKTHIST occurrences. If any more of these occurrences were found, the report would be inaccurate as to the contents of the database, since the sets would appear to be empty. Statements to reset the HLAST, ELAST, and MORE flags were included to obtain a report that reflected the true contents of the implemented database.

5. REC_FIND Procedure.

This procedure locates the actual record requested by the user. It did not handle record additions well because of currency constraints imposed by IDMS.

IDMS is a network oriented DBMS; the way it handles record requests is by navigating through the sets (except with CALC records or direct records, which can be located directly using a database key). The last record called by the user through the OBTAIN command is the one selected as the "current of run-unit", i.e., the one the user can work with. Because of this characteristic, the procedure had to be changed drastically to respond to the needs of a larger database.

The new REC_FIND procedure can now do the following correctly:

a. Handle new product additions. Statements were written to allow for the setting of the area, region and record type currencies to add a new product. Birchard's [5] version did not handle new product additions.

b. Establish correct currencies for MARKET, MKTEST, and MKTHIST records in order to add, change or get records. Due to the relatively larger database used in the implementation Birchard's [5] algorithm was incorrect. For example, if a user requested a MKTEST record for the wrong PRODUCT, the program found the MARKET owner record (using its CALC key) of the set EST for another PRODUCT and performed the commands on the wrong record. In the new subroutine currencies are checked as to set ownerships before commands are carried out. If the ownership is incorrect, an error message is produced and control passes to the main program.

6. GIT Procedure.

This procedure isolates keywords in the command typed by the user and assigns those keywords to program variables. Statements setting the DO_CMD flag were added to insure proper processing of commands by the PROC_CMD procedure; without these statements, PROC_CMD would produce misleading messages (i.e. say a record had been changed or added when in reality that had not happened due to errors in the command syntax.)

7. DBPARSE_CMD Procedure.

This procedure parses the command entered by the user through iterative calls to the GIT procedure.

A new segment of code was created for this procedure to handle record additions. This was necessary because the syntax of the DBADD command was changed to make it more logical to the user. The DBADD command's syntax in Birchard's [5] version was as follows:

```
DBADD^after^<ARGSET>
```

```
ARGSET::=<RECORD_ID>^<RECORD_TYPE>|
        <RECORD_ID>^<RECORD_TYPE>^for^<PRODNAME>|
        <RECORD_ID>^<RECORD_TYPE>^for^<PRODNAME>^in^the^
<MKTNAME>
^::= one or more blanks
RECORD_TYPE::= PRODUCT| MARKET | MKTHIST | MKTEST |DEVHIST
RECORD_ID::= PRODNAME | MKTNAME | FDATE | PERIOD
```

(This syntax has been derived from his syntax description and the DBMENU instructions, since neither one was totally accurate).

This syntax was definitely awkward for a person who is not involved in the actual design of the database, such as the user, who really does not care if the record is actually added "after" something. The new syntax is as follows:

```
DBADD^a^<ARGSET>
^::= one or more blanks
ARGSET::= <RECORD_TYPE>|
            <RECORD_TYPE>^for^<PRODNAME>|
            <RECORD_TYPE>^for^<PRODNAME>^in^the^<MKTKNAME>
RECORD_TYPE::= NEW_PRODUCT | MARKET | MKTHIST | MKTEST
```

Furthermore, statements were added to this procedure to advise users of possible errors found in command syntax.

8. DBADD Procedure

This procedure is invoked when the user wants to add a new record occurrence to the database.

The same type of statements created for DBCHANGE were added to this procedure to handle the situation in which the user keys in the wrong command and does not wish to proceed further with record additions. In Birchard's version no provision was made for this contingency and IDMS would, again, respond with an error message and cancellation of the session.

9. PROC_CMD Procedure

This procedure determines which command will be executed (DBADD, DBDELETE, DBCHANGE, DBGET, DBREPORT, LEAVE) after the input string from the user is parsed by the DBPARSE_CMD subroutine. The following changes were made to insure proper processing:

The DO_CMD flag is checked to insure proper processing of commands.

The section of this procedure corresponding to the DBADD command was totally changed to handle:

a) New product additions (as mentioned before, this was not possible in the original version of the front-end).

b) Initialization of field values for the new record to blanks. In Birchard's [5] version IDMS would use the "current" record as a template and change only those values entered by the user; if the user did not fill in all the fields with new values, when the new record was stored it would contain the values from the "template" record in the other fields.

The CHG_OK variable is used to determine if the message confirming the record addition should be sent back to the user.

The performance of the front-end with these changes improved greatly. Error messages are more precise and it is assured that IDMS gets the proper information before processing commands.

4.3 Evaluation of Tool Performance

Having described the actual development of the implemented software tools as well as the goals of the lectures, cases, and guest lecturers included as part of the classroom course implementation, Chapter V will proceed with the examination of the appropriateness of the aforementioned tools, and evaluate their success according to the results of course evaluations conducted both by the students and the instructor.

CHAPTER V

EVALUATION OF COURSE CONTENT

5.1 Instructor's Course Evaluation

The following is an evaluation of the course content from the instructor's viewpoint. The results of the evaluation conducted by the students is found in section 5.2 of this chapter.

5.1.1 Course Design Evaluation

It is felt that the previous work by Strunk [4] and Birchard [5] was geared in the right direction and that it helped as the basis for the development of some of the objectives and tools employed during the first semester of course implementation. The correctness of the approach developed independenly here and used so far for this course is made more evident when it is compared to the trends found for similar courses in universities around the nation. The following information concerning those trends was found in a report by Dr. Raymond McLeod, Jr. entitled "The Undergraduate MIS course at A.A.C.S.B. Schools: A Special Report to Participating Deans and Professors" [6], which came to the attention of this author after most work on this thesis had been completed.

Dr. McLeod conducted a survey among American Assembly of Collegiate Schools of Business (A.A.C.S.B.) members regarding the teaching of undergraduate MIS courses. Some of his findings are especially interesting for universities like Kansas State

University, which are just beginning to implement the first core course in this field. The following are some of his findings:

1. In all responding schools, the MIS course is being taught within the school of business.
2. The courses covered a wide range of topics, but the main ones are listed in the following table:

Table 1

Topics Included in the MIS Course

Topic		Number of Schools	Percent
1.	Systems Analysis and Design	55	88.7
2.	Systems Theory	53	85.5
3.	Computing equipment	52	83.9
4.	Data Base	51	82.3
5.	MIS by Management Level	49	79.0
6.	System Life Cycle	49	79.0
7.	Decision Support Systems	46	74.2
8.	Data Base Management Systems	45	72.6
9.	MIS by functional area	45	72.6
10.	Programming Language(s)	43	69.4
11.	Computer Security	40	64.5
12.	Management theory	33	53.2
13.	Data Communications	29	46.9
14.	Other	18	29.0

3. Professors are using the MIS course to teach systems analysis and design.
4. The MIS course has an emphasis on computers rather than on management. Dr. McLeod considers this to be an unfortunate fact, since many of the universities did not require the basic management course as a prerequisite, this tends to indicate that the students at many universities are studying MIS without a solid foundation in management.
5. The fact that computer hardware is ranked so high confirms the suspicion that such hardware is not covered adequately in the introduction to computing course.
6. Out of the experiential activities found in the various universities, the following were the most common: non-computer based case problems, computer based case problems, computer based management decision games and, surprisingly,

programming in one of the more traditional languages, such as BASIC, COBOL, and FORTRAN. Students also get hands-on computing experience through the use of statistical packages. Very popular are other pre-written software packages, i.e., linear programming. The electronic spreadsheets such as VisiCalc, SuperCalc, Multiplan, and IFPS receive strong support as do word processing packages such as Wordstar and Easywriter. Dr. McLeod found that teaching of programming in the MIS course defied logic, since it is not the place to sharpen programming skills: the key to a successful MIS, he feels, is the joint work of the manager and the systems analyst in problem definition and solution formulation; therefore, that is the topic that he considers should receive course emphasis.

7. Although 83.2 percent of the universities include data base as a topic in the MIS course, less than one fourth provide an opportunity for hands-on use of database management systems software. Of the packages being used, none is an overwhelming favorite, although dBaseII was mentioned most often. Two reasons for this are (1) the time required to develop an ability to use a DBMS, and (2) the hardware and software resources required.

8. The experiential activity that can take the most time to discharge effectively and has the greatest potential in terms of conveying the essence of MIS design and operation, is the term project in a real organization.

9. The key to success for the course mentioned most frequently was the term project, followed closely by hands-on experience and cases.

10. The MIS course appears to be heading in the future towards the use of smaller scale hardware systems (mini-micro systems) and their accompanying software. Also, more emphasis will be given to the DSS, end-user computing, and fourth generation languages. [6]

In looking at this review it is felt that the present approach at Kansas State University already meets the requirements of emphasis on problem definition and solution formulation. There are some aspects in which we might have an advantage,such as:

1. Requirement of the management course as a prerequisite for the class.

2. Including hands-on experience with a database management system.

3. Having a strong management orientation instead of a hardware or programming orientation to the MIS.

The aspects in which we might have a disadvantage are:

1. Lack of a term project in a real organization.

2. Unavailability of small scale hardware systems for software implementation.

5.1.2. Evaluation of Previous Work

As mentioned in the previous section, Strunk's [4] and Birchard's [5] work has been very helpful in building the format and content of the course. Neil Strunk [4] provided a series of potential SBLO (these SBLO can be found in Appendix D) and Birchard developed some of these SBLO (DBMS and DSS) to the point of making them potentially usable for actual classroom implementation.

Strunk's [4] SBLO are specific and clear as to what needs to be done to accomplish them. These SBLO were also found to be consistent with what Dr. McLeod found is being taught at other colleges around the nation in the core MIS course.

However, Birchard's [5] work needed some changes (as specified in Chapter IV) in order to make a more realistic implementation possible. If it had been necessary to develop a new IDMS database for class use (judging by the time it took to update the front-end program) the SBLO for DBMS would probably have not been ready for this semester's classroom implementation.

However, there are some deep rooted problems with Birchard's [5] schema design. Figure 1 shows a Data Structure Diagram for the schema. In his design, Birchard tried to build the records so that the user could directly enter values found in EMPIRE Model 2 into the record fields. That might be the reason his records and sets are not very realistic when examined from a business organization's point of view. For example, by looking at the data structure diagram, it can be seen that a PRODUCT record can own several MARKET records but a MARKET record cannot own several product records. This is very awkward, especially since the MARKET record's CALC key is the MKTNAME and duplicates are not allowed; therefore, the user cannot specify that two or more products are to be sold in the same market which is the usual cases in real life situations. Also, the MKTEST record contains information, such as the Earnings per Share (EARN), that would really apply to all the markets where the company sells the product and not be different for each specific market. Another problem with this schema design is that it is very specifically matched to the EMPIRE Model 2: the database cannot be used for anything else but assignments derived from that model or facsimiles of it.

The front-end program which simulates the query language, it works well after undergoing the changes mentioned in chapter IV. However, and this is unavoidable since it was designed for the present schema, it is also very specific to the particular implementation and cannot be used with another schema design.

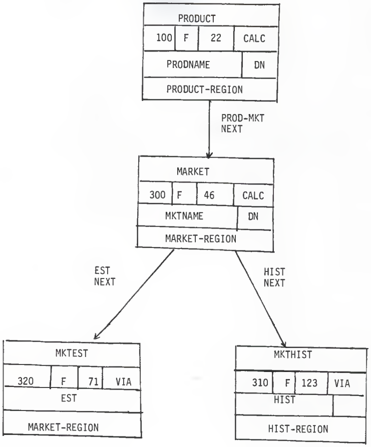

Figure 1: Schema Data Structure Diagram

The following sections cover the other parts of the course content (text editing tool, case analyses, and lectures) that were developed for the classroom implementation during spring semester, 1984.

5.1.3 Evaluation of Text Editing Tool

SCRIPT is by no means the ideal software for word processing. It is cumbersome to use and the inexperienced user has a hard time coordinating the input formatting commands and the way the final output is supposed to look (i.e., what you see is not what you get); but since it is the word processing tool available through the mainframe computer at KSU it provided the only chance for most of the students to get hands-on experience in this area. It also helped to give them more confidence when using interactive computer systems.

5.1.4 Case Analyses Evaluation

The case analyses included as part of the course were very valuable in that they helped provide the management emphasis for the course. Students were placed in a position to study realistic MIS related situations and work in teams to reach decisions to solve the problems found in those situations.

5.1.5 Lectures and Guest Lecturers

The lectures emphasize the areas deemed relevant given the goals of the course (a Class Syllabus is found in Appendix D). This part of the course was very important in comunicating the main concepts and definitions necessary to understand the functioning of an MIS in an organizational environment. The guest lecturers helped convey to the students what happens when these concepts are applied to real computerized systems (i.e., problems with scheduling high priority jobs with limited computer resources, factors considered when implementing networks, etc.)

5.2. Student Evaluation of Course Content

An evaluation was conducted after all the software assignments and case analyses were handed in by the students. A sample of the survey form is found in Appendix D.

This form was built using Likert scales, sometimes referred to as Likert-type scales or summated scales, which are widely used in marketing research.

A Likert scale requires a respondent to indicate a degree of agreement or disagreement with each of a series of items, generally statements, related to the attitude object. These levels of agreement-disagreement are then scored in such a way as to consistently reflect positive and negative attitudes. [7]

In the present case, the scales were built in such a way that a high score indicates that the student agrees that particular statement is correct. For example, for the following evaluation question on the EMPIRE assignment (Figure 2) the values for each choice were assigned from a low of -2 to a high of 2, with the high scores indicating that the student felt strongly that the statement adequately described a characteristic of that specific assignment; obviously, a score of -2 indicates strong disagreement with the statement.

Figure 2:

Sample of student evaluation question

The Empire assignment helped you to:

1. Understand how Decision Support Systems (DSS) work.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
(2)	(1)	(0)	(-1)	(-2)

(A copy of the complete form can be found in Appendix D.)

In this way it was possible to get ordinal data that would allow the instructor to make inferences about how well the students considered that the assignments fulfilled the established SBLO. These numerical values were not included in the evaluation form given to the students.

<u>5.2.1 Results of Student Evaluation</u>

The questions in the evaluation form were written to cover each one of the SBLO established for the course, as well as the lectures and documentation provided for the assignments.

There were 22 students present the day the evaluation was conducted (out of an enrollment of 27). The answers were tabulated using the scalar weights described in the previous section and Table 2 summarizes the results. The column for each possible choice is headed by the scalar value assigned to it. The Total column is the result of adding across each row the values obtained by multiplying the number of answers for each choice times its scalar weight. This provided the instructor

Table 3

Summary of Results of Student Evaluation

	Number of Students selecting each Choice					
Choice number	1	2	3	4	5	Weighted
Scalar Weights	(2)	(1)	(0)	(-1)	(-2)	Total
QUESTIONS						
Script						
The Script assignment helped you to:						
1. Learn how to sign on to the mainframe computer at KSU using CMS	16	5	1	-	-	37
2. Learn how to use the Xedit editing environment	6	14	2	-	-	26
3. Gain confidence with interactive systems	4	16	2	-	-	24
4. Learn how to obtain typed output at remote laboratory	11	8	2	1	-	29
5. Learn Script commands to produce desired format for output	7	11	4	-	-	25
Total for Text-Editing SBLO						141
Empire						
The Empire assignments helped you to:						
1. Understand how Decision Support Systems (DSS) work	5	13	2	2	-	21
2. Analyze problem data to produce valuable information	4	12	4	2	-	18
3. Understand how a decision may be improved by using DSS	9	10	3	-	-	27
4. Increase your confidence in working with interactive systems	6	12	4	-	-	24
5. Learn how to interpret output from DSS tools	3	15	4	-	-	21
Total for Empire DSS SBLO						111
IDMS						
The IDMS assignment helped you to:						
1. Learn how a query language works	8	14	-	-	-	30
2. Learn to interact with a Database Management System	7	15	-	-	-	29

Table 3 (Continued)

Summary of Results of Student Evaluation

	Number of Students selecting each Choice					
Choice number	1	2	3	4	5	Weighted
Scalar weights	(2)	(1)	(0)	(-1)	(-2)	Total
IDMS (cont'd)						
3. Learn how to integrate DBMS and DSS tools (i.e., see the database is updated to reflect the latest decicions made by management	9	10	3	-	-	28
4. Learn how to add, delete, change, and retrieve records in a database	11	11	-	-	-	33
5. Learn how to produce reports reflecting your DBMS transactions	7	13	2	-	-	27
Total for DBMS SBLO						148
LECTURES						
Lectures helped you to understand:						
1. MIS and DSS elements, concepts, and functions	5	15	2	-	-	25
2. General Systems and management concepts	6	10	6	-	-	22
3. Computerized systems concepts	8	11	3	-	-	27
4. Planning, analysis and design, implementation, and operation of an MIS	6	8	7	1	-	19
5. Use of an MIS at different levels of management	4	14	4	-	-	22
Total for Lectures						115
DOCUMENTATION						
The documentation and instructions given for the assignments:						
1. Made assignments easy to fulfill	10	10	1	1	-	29
2. Were thorough and descriptive	9	11	2	-	-	29

with a quantitative value to judge the opinion of the students on how successful the assignment was in achieving the SBLO.

As these summary results show, the majority of the students felt they were able to achieve most of the SBLO to a satisfactory degree. The SBLO that seem to have had the highest level of success, judging from the cumulative scores, are the ones applicable to the IDMS DBMS assignment (with a total score of 148) and text editing with SCRIPT (with a total score of 141).

Apparently, from the comments and suggestions made to improve the class in future semesters, it seems the students consider that they would learn much more out of the EMPIRE assignments if they were taught about the system itself and allowed to write their own models for implementation.

Other suggestions included the scheduling of more guest lecturers who could talk about actual database implementations, and more computer assignments during the semester (i.e., using spreadsheets and other software packages). This last suggestion is encouraging, since it seems to indicate they feel much more at ease with interactive systems and wish they had had more opportunities to use them during the semester. It was also suggested that readings from current literature should be included as part of the course content to be able to keep abreast of the latest developments since this is such a rapidly changing field.

5.3 Evaluation Analysis

Overall, the ratings given by the students seem to be satisfactory as to the accomplishment of class objectives for this spring semester, 1984 and the instructor's evaluation of course content seem to indicate that the course adequately met the needs of the curriculum and the SBLO selected for the fulfillment of those needs. In the following chapter, conclusions will be drawn as to what was accomplished this semester in this course implementation. Also, the possible near-future developments for the MIS course at the College of Business will be discussed.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

This thesis has described the work conducted to achieve a successful classroom implementation (including the refinement and further development) of the SBLO designed by Neil Strunk [4] and the software tools designed by Robert Birchard [5]. It also has given an evaluation of the degree of achievement of those SBLO both from the author's and the students' point of view. The following conclusions and recommendations have beem drawn from the experience gained from these implementation and evaluation.

6.1. Student Behavioral Learning Objectives (SBLO)

The student behavioral objectives will need to be expanded to cover at least all the areas developed by Strunk [4] (as described in Appendix D): DBMS, DSS, Office Automation, and Transaction Processing. The pressure of advancing technology in the field of computers and software make it imperative for the students to get an adequate preparation in college to be able to operate efficiently in an organizational environment upon graduation.

To make it possible for the students to fulfill these SBLO effectively, additional computer equipment will have to be acquired by the College of Business. At present, upper level accounting and finance students have access to micro-computer labs (equipped with Zenith 100s and IBM PCs) to work with spreadsheets and financial data. The Marketing Department will be installing next fall (Fall 1984), an NCR Tower system with UNIX operating system, 2 megabytes main memory, 80 megabytes hard disk, and 16 terminals. The Management Department has plans to install during spring semester 1985 a micro-computer lab with approximately 14 Zenith 150s to be used by the students taking the MIS and other quantitative courses in the department. This lab will offer the students in the MIS course the opportunity to work with word-processors (Word Perfect), spreadsheets (Lotus 1,2,3, and Multiplan), statistical packages (Statpro and Abstat), microcomputer database management systems (dBaseII) and, hopefully, a calendar system. When this lab is in operation, the SBLO can be expanded to have the students write their own database applications using a system that is more "user friendly" than IDMS; also, the available software will permit the use of other tools (such as spreadsheets and statistics packages) for financial problem solving and strategic analysis. Until this time comes the SBLO will have to be expanded, subject to the limitations of the mainframe computer at KSU. These planned expansions and modifications are delineated below.

6.1.1. IDMS SBLO

A new real-life database will be installed for classroom use in Fall semester, 1984. This database will use data from a local organization making it possible to have a more realistic approach to the implementation of the SBLO for the database management systems.

A new front-end will also be needed since the KSU Computer Services does not support the query language available for IDMS. The students will be required to interact with the database throughout the semester so they will be able to become more familiarized with the basic database manipulation commands.

6.1.2. EMPIRE SBLO

The EMPIRE system will be used, but students will be required to learn enough about the system itself to be able to write their own models for implementation. These models will be based on the information needed to interact with the new database and to complete case analyses given during the class.

6.3. Long-term outlook

If the computer facilities planned for the College of Business in general and the Management Department in particular become a reality, we would be able to follow the trend in the teaching of the core MIS course around the nation, as discussed by Dr. McLeod [6] in his paper. In his opinion:

> "It is an absolute necessity that today's and tomorrow's college graduates know how to use the computer as a decision support system--to generate information for decision making. The industry trend is toward end-user computing where the manager can use the computer independently of the system analyst and programmer. The MIS course and integration of MIS concepts into the entire business curriculum represent the vehicles for achieving this MIS literacy.
>
> Within the school of business, the dean must initiate programs contributing to improved computer skills among the faculty. The MIS faculty can participate by conducting seminars, demonstrations, hands-on laboratory sessions, and so on. The roles played by the A.A.C.S.B. and the MIS faculty are important. But, in the long run, the degree to which students learn to use the computer as a problem-solving tool will be largely determined by the abilities of the deans in assembling a

> faculty skilled in integrating modern computing hardware and software experiences into the business curriculum." [6]

The acquisition of the right facilities will definitely be a step in the right direction in order to fully satisfy the needs of the students and the curriculum.

APPENDIX A

STUDENT MANUAL FOR SCRIPT ASSIGNMENT

The following is the sequence of tasks that you must perform in order to sign on to the computer and create a script file:

(These instructions apply to the terminals in Calvin Hall, Room 9)

1) Turn on terminal (switch is towards the back, right-hand side corner).
2) When terminal is ready (warm), type as follows:

L VMxxx (where xxx stands for the last three digits or letters of your account number)

THE COMPUTER WILL REPLY:

ACCOUNT NUMBER

(TYPE YOUR ACCOUNT NUMBER, HIT RETURN KEY)

THE COMPUTER WILL REPLY:

ACCOUNT PASSWORD

(type in your account password)

THE COMPUTER WILL REPLY:

SOCIAL SECURITY NUMBER

(type in your social security number)

This procedure should get you signed on to the computer.

The first thing I want you to do is to create a PROFILE EXEC. To do that, follow these instructions:

1) Type XEDIT PROFILE EXEC

The computer will reply: Creating new file.

2) Type INPUT

The computer will reply: Input mode:

3) Type CP TERM ESCAPE |
Hit return key
Type CP TERM CHARDEL

4) Hit return key twice to get out of input mode.

5) Type CH/CHARDEL/CHARDEL<hit space bar><hit backspace key>
The computer will answer by saying: one occurrence changed.

6) Type FILE
(This will put your profile exec in you A disk).

What you have done, is to create a method to be able to use the backspace key to correct your typos, and to be able to use the " within you text.

Now, I want you to create a script file. To do that, all you have to do is:

1) Type XEDIT FN SCRIPT
Where FN is the name you want to give your file (should be alphabetic, 8 characters or less).
Again, the computer will reply, CREATING NEW FILE:.
Type INPUT
The computer will reply, INPUT MODE:
Type in your file, following the commands I am giving you in the example.
I want you to type at least 2 pages of input, and to use at least the commands that I am giving in the example so you get some hands-on experience with the terminal.

There are some commands in the xedit editor that will be useful to you:
ch/old/new (TO CHANGE CHARACTER old STRINGS INTO new STRINGS IN YOUR TEXT)
l/string (TO LOCATE CHARACTER STRING IN YOUR TEXT)
down n (TO GO DOWN IN YOUR TEXT n LINES)
up n (TO GO n LINES UP IN YOUR TEXT)
top (TO GO TO THE TOP OF YOUR FILE)
bottom (TO GO TO THE BOTTOM OF YOUR FILE)

All of these commands will take effect when done outside of the input mode, so make sure that when you want to use them you are out of input mode.
To get out of input mode, just hit the return key twice, and you get back into xedit mode.
Once you are done with your editing, type
FILE
To store your terminal session. This will be stored in you filemanager space, and you can retrieve it later by typing XEDIT FN SCRIPT whenever you want to look at it. (Again, FN stands for your file name.)

If you want to see how your scripted file looks, the best thing to do is to type:

```
SCRIPT FN (DISK
```

Where FN stands for your file name.

This will script your file into your disk. To look at it, type

```
XEDIT FN LISTING
```

Again, FN is your file name.

then type:

```
t 25
```

This will tell the computer that you want the next 25 lines typed.
When you have viewed them, you can type

```
t 25
```

Again and view your entire file that way.

This LISTING file will have the codes for the printer, but the general format should be what you should be interested in.
Once you are sure that your output looks like you want it to, then type:

```
CP SPOOL CONSOLE TO * START
```

then type:

```
TYPE FN SCRIPT
```

(When this is ready, then type:)

```
SCRIPT FN
```

(When this is ready, then type:)

```
CP SPOOL CONSOLE CLOSE STOP
```

(when this is ready, then type:

```
READ FN2 FT
```

(Where FN2 is another file name and FT is another file type alphabetic, 8 characters or less).

When you have done this, then type:

```
OSPRINT FN2 FT
```

(Then copy down the VM number the system will give you, go to Cardwell and claim your output at whe window).
The following is the example for this assignment.

```
.tt /Example of Script for Students in MIS Class//pn %/
.BX 10 60
.IN 5
.SK
.CE SOFTWARE SELECTION
.CE It's not a game
.sk
.bx off
.sk 3
.pp
The purpose of this article is to provide a potential user with a
recommendation for how to choose among competing software
packages.
It will provide users with the framework and the techniques for
the evaluation of various competing systems currently on the
market
it assumes the decision
```

to acquire a system has been made. While one might also give
some thought to the
possibility of in-house development of a major software system,
the investment
and maintenance costs of such a system are usually prohibitive.
(For a brief
comparison of the in-house vs. outside purchase, refer to "why
choose an Accounting
Software Package?," Steven J. Weinberg, Management Accouting,
February, 1980.)
.sk 3
.us Determination of Needs
.sk 2
.pp
as in any project someone has to be given responsibility and
authority for its
timely completion.
Some organization have a specific department which will fulfill
this need, others
require the appointment of an individual with the appropriate
supporting personnel.
In this article we will call the designated individual the
project leader (PL).
.sk 2
.pp
Having selected the individual who will be responsible for the
project, it must now
be determined what specifically the PL will be responsible for.
That is,
we must now determine the user needs. As an initial starting
point in the determination
of the users' needs, we must first determine who the users are,
the proceed to query those users to determine their specific
needs. Using a fixed asset system as
an example, the following areas of responsibility would be the
prime users: financial
accounting, property management, operational accounting, tax
accounting, and those
individual responsible for reporting to a specific regulatory
body. If a manual system
is presently being employed, a review of the users of that system
would be a
logical starting point.
.sk 2
.pp
Having determined present and potential users, we must obtain
from them at least a
cursory listing of their requirements. Usually, this is done by
the PL sending a memo to users requesting that they
furnish the PL with a written schedule
depicting the users' requirements, their rationale for the
request and,
when possible, a pro-forma example of the report output. In the
request, the PL
might suggest that each user review his or her present system for
those reports which
should be retained, modified or eliminated.

```
.sk 2
.pp
The resulting feedback will give the PL an overview of the
general requirements of the
firm allowing the PL to proceed to the second major step--a
general screening of vendors.
.sk 3
.us General Screening techniques
.sk 2
The general screening of most systems is similar.  The primary
purpose of
the general screening is to pare down the list of potential
vendors from
approximately 10 to a manageable three to five.  The PL is
responsible
for this refinement of the selection list which should ultimately
come down to those vendors which can demonstrate credibility to
both the PL and
the PL's staff.
.pp
The general screening of vendors is a two-phase procedure.  Phase
one is the general
solicitation of vendors through a written memo describing the
user's general
needs as determined in the previous section of this article.  The
memo should
state
.in 5
.hi 4
1)  Who the user is
.br
2)  The type of hardware
.br
3)  What general attributes must be in the package:  i.e., for
fixed
assets; e.g., The Financial Accounting Standards Board (FASB)
Nos. 8, 13,
33;  Accelerated Cost Recovery System (ACRS) capabiities, and
information
concerning the general price range.
.br
.in
.hi
.pp
The second phase of the screening process concerns itself with an
evaluation of those vendors who reply to the initial inquiry.
Some
general credibility criteria are as follows:
.ce
.us References
.pp
How many references may each vendor furnish and in what
industries
are the references doing business  While the number of references
is
not a sufficient criterion for credibility, it is indicative of
the
```

general acceptability of the system.

THIS ARE THE INSTRUCTIONS TO SCRIPT THE FILE:

```
script test
Waterloo SCRIPT - Version 82.1 (82OCT18)
Load paper for page 1; hit return:
```

Example of Script for Students in MIS Class pn 1

```
+-----------------------------------------------+
^                                               ^
^              SOFTWARE SELECTION               ^
^               It's not a game                 ^
^                                               ^
+-----------------------------------------------+
```

The purpose of this article is to provide a potential user with a recommendation for how to choose among competing software packages. It will provide users with the framework and the techniques for the evaluation of various competing systems currently on the market. it assumes the decision to acquire a system has been made. While one might also give some thought to the possibility of in-house development of a major software system, the investment and maintenance costs of such a system are usually prohibitive. (For a brief comparison of the in-house vs. outside purchase, refer to "why choose an Accounting Software Package?," Steven J. Weinberg, Management Accouting, February, 1980.)

Determination of Needs

as in any project someone has to be given responsibility and authority for its timely completion. Some organization have a specific department which will fulfill this need, others require the appointment of an individual with the appropriate supporting personnel. In this article we will call the designated individual the project leader (PL).

Having selected the individual who will be responsible for the project, it must now be determined what specifically the PL will be responsible for. That is, we must now determine the user needs. As an initial starting point in the determination of the users' needs, we must first determine who the users are, the proceed to query those users to determine their specific needs. Using a fixed asset system as an example, the following areas of responsibility would be the prime users: financial accounting, property management, operational accounting, tax accounting, and those individual responsible for reporting to a specific regulatory body. If a manual system is presently being employed, a review of the users of that system would be a logical starting point.

Having determined present and potential users, we must obtain from them at least a cursory listing of their requirements. Usually, this is done by the PL sending a memo to users requesting that they furnish the PL with a written schedule depicting the users' requirements, their rationale for the request and, when possible, a pro-forma example of the report output. In the request, the PL might suggest that each user review his or her present system for those reports which should be retained, modified or eliminated.

The resulting feedback will give the PL an overview of the general requirements of the firm allowing the PL to proceed to the second major step--a general screening of vendors.

General Screening Techniques

The general screening of most systems is similar. The primary purpose of the general screening is to pare down the list of potential vendors from approximately 10 to a manageable three to five. The PL is responsible for this refinement of the selection list which should ultimately come down to those vendors which can demonstrate credibility to both the PL and the PL's staff.

The general screening of vendors is a two-phase procedure. Phase one is the general solicitation of vendors through a written memo describing the user's general needs as determined in the previous section of this article. The memo should state

1) Who the user is
2) The type of hardware
3) What general attributes must be in the package: i.e., for fixed assets; e.g., The Financial Accounting Standards Board (FASB) Nos. 8, 13, 33; Accelerated Cost Recovery System (ACRS) capabiities, and information concerning the general price range.

The second phase of the screening process concerns itself with an evaluation of those vendors who reply to the initial inquiry. Some general credibility criteria are as follows:

References

How many references may each vendor furnish and in what industries are the references doing business While the number of references is not a sufficient criterion for credibility, it is indicative of the general acceptability of the system.

```
*****************************
PLEASE DO NOT FORGET TO TYPE:
          LOG
 WHEN YOU ARE DONE WITH YOUR ENTIRE TERMINAL SESSION
******************************
```

APPENDIX B

MIS - EMPIRE ASSIGNMENT

Instructions

Today you are getting two handouts with this instruction sheet.

HANDOUT ONE

This handout contains the listing of all the files used for your actual empire runs.

You must go to either Calvin Hall, room 9, or Cardwell, room 23, or Farrell Library (reserve desk where I have put two copies of the Empire Manual) and look up in the empire introduction manual what each command line in model2 is doing. Write the description of what the command does next to the corresponding line in the handout.

HANDOUT TWO

This handout contains the actual session record from running the following models through Empire: FINANSUM, MODEL2, MODEL5.

I want you to follow the directions in that handout to run the models on your own. You do not have to copy the listing files; they are already in your A disk. Just follow the instructions in the handout.

POINTS

The completion of the tasks specified here will constitute completion of assignment for a total of 25 points. If you want to get the total possible points (30), you must go to either Calvin Hall, room 9, of Cardwell, room 23, and look up what the plot, simulate, what is, and what impacts commands do. This commands are found in handout two or in your printout after running the models. Write what each one of this commands is doing in the appropriate line. (The description of the commands is found in the Empire "decision support system", user Reference Manual. This manual is not in Farrell Library, only in Calvin room 9 and Cardwell room 23.)

HANDOUT ONE

```
TYPE FINANSUM EMPMOD

OPTION SECTION
CONTROL FSUM
ROWTITLE 100
COLUMN 14
WIDTH 132
COLUMN SECTION
A1970 "1970"
A1971 "1971"
A1972 "1972"
A1973 "1973"
A1974 "1974"
A1975 "1975"
A1976 "1976"
A1977 "1977"
A1978 "1978"
A1979 "1979"
;SUMMARY DATA
S7074 "SUBTOTAL FOR/1970-1974"
S7579 "SUBTOTAL FOR/1975-1979"
T7079 "TOTAL FOR/1970-1979"
ROW SECTION
;EARNINGS STATISTICS
NSALES "NET SALES................................"
ECOSTS "EMPLOYMENT COSTS........................."
MATSER "MATERIALS AND SERVICE...................."
DEP    "DEPRECIATION............................."
TAXES  "INCOME TAXES............................."
TCOSTS "                                        "
OPINC  "OPERATING INCOME........................."
INTINC "INTEREST, DIVIDENDS AND OTHER INCOME......"
INTEXP "INTEREST AND OTHER DEBT CHARGES..........."
CLCOST "ESTIMATED CLOSEDOWN COST................."
FLDEXP "FLOOD EXPENSE............................"
INCBTX "INCOME(LOSS) BEFORE INCOME TAXES.........."
INCTAX "INCOME TAXES............................."
NETINC "AMOUNT..................................."
RULES SECTION
;COMPUTE INCOME STATEMENT RELATIONSHIPS
FOR COL=A1970 TO A1979 DO
     TCOSTS=COLSUM(ECOSTS,TAXES,COL)
     OPINC=NSALES-TCOSTS
     INCBTX=OPINC+COLSUM(INTINC,FLDEXP,COL)
     NETINC=INCBTX-INCTAX
END
;SUMMARY OF PERFORMANCE
FOR ROW=NSALES TO NETINC DO
     A(S7074,ROW)=SUM(A(1,ROW),A1970,A1974)
     A(S7579,ROW)=SUM(A(1,ROW),A1975,A1979)
     A(T7079,ROW)=SUM(A(1,ROW),S7074,S7579)
END

R;
T FSUM EMPCON

DATA
OPEN FSUMDATA
READ NSALES
READ ECOSTS
READ MATSER
READ DEP
READ TAXES
READ INTINC
READ INTEXP
READ CLCOST
READ FLDEXP
READ INCTAX
CLOSE
END
RUN FINANSUM
PRINT FROM REP

R;
T FSUMDATA EMPDATA

NSALES 2935.4,2969.1,3113.6,4137.6,5381.0,4977.2,5248.0,5370.0 &
6184.9,7137.2
ECOSTS 1361.7,1323.6,1448.6,1759.3,2072.0,2193.2,2313.6,2368.5 &
2550.0,2939.1
MATSER 1214.4,1198.5,1208.4,1765.1,2442.7,2240.4,2373.0,2707.0 &
2889.8,3367.7
DEP 165.0,159.3,180.8,196.1,210.9,234.2,275.6,300.1,321.9,351.3
TAXES 54.6,60.7,60.5,58.8,63.1,68.1,70.8,72.3,74.4,86.1
INTINC 28.8,32.5,26.6,41.3,67.7,51.1,56.7,40.2,47.7,81.7
INTEXP -33.4,-38.3,-40.3,-43.0,-44.0,-63.4,-77.7,-82.5,-86.4 &
-80.0
CLCOST 7*0,-750
FLDEXP(8) -41.0
INCTAX 44.0,82.0,67.0,150.0,274.0,41.0,26.0,-463.0,85.0,119.0

R;
T REP EMPREP

PAGE 60,6
WIDTH 132
COLUMN 14
ROWTITLE 60
HEADING CENTER
EXCHANGE 0 "   -   "
MARKS OFF
SELECT S7074:T7079
TITLE 1 LEFT "JUSTIN ALLEN PETROLEUM CORP AND SUBSIDIARY COMPANIES"
TITLE 2 LEFT "TEN YEAR FINANCIAL SUMMARY"//
```

```
SUBTITLE "(DOLLARS IN MILLIONS)"
SUBTITLE "EARNINGS STATISTICS"
SUFFIX
DECIMAL 1
PRINT NSALES
SUBTITLE "COSTS AND EXPENSES:"
SUBTITLE "OPERATING CHARGES:"
PRINT ECOSTS
PREFIX
PRINT MATSER:DEP
SUBTITLE " TAXES, OTHER THAN EMPLOYMENT AND"
PRINT TAXES=
PREFIX "$"
PRINT TCOSTS=OPINC
PREFIX
SUBTITLE "OTHER INCOME(EXPENSE)"
PRINT INTINC:FLDEXP=
PREFIX "$"
PRINT INCBTX
PREFIX
PRINT INCTAX=
SUBTITLE "NET INCOME (LOSS)"
PREFIX "$"
PRINT NETINC

R;
T MODEL2 EMPMOD

OPTION SECTION
CONTROL CONTROL2
COLUMN SECTION
CREATE QTR 1:20 "HISTORY"
MAR     "PERIOD/ENDING/3-31"
JUN     "PERIOD/ENDING/6-30"
SEP     "PERIOD/ENDING/9-30"
DEC     "PERIOD/ENDING/12-31"
TOTAL  "TOTAL/FOR/YEAR"
ROW SECTION
MKTSHR INPUT "MARKET SHARE"
TUNITS     "TOTAL MARKET"
UNITS      "UNITS SOLD"
SALES      "SALES REVENUE"
COS        "COST OF SALES"
GPROF      "GROSS PROFIT"
OVHD       "OVERHEAD EXPENSE" 5000*
PBT        "PROFIT BEFORE TAX"
SCALAR SECTION
PRICE INPUT "UNIT PRICE"
UCOST INPUT "UNIT COST"
TAXRAT "TAX RATE" .48
TAX        "INCOME TAX"
PAT        "PROFIT AFTER TAX"
EARN       "EARNINGS PER SHARE"
RULES SECTION
```

```
FOR COL=MAR TO DEC DO
   UNITS=ROUND(MKTSHR(COL)*TUNITS(COL),0)
   SALES=UNITS*PRICE
   COS=UNITS*UCOST
   GPROF=SALES-COS
   IF COL NE MAR THEN DO
        IF SALES GT SALES(COL-1) THEN &
           OVHD=OVHD(COL-1)+.05*(SALES-SALES(COL-1))
        ELSE OVHD=OVHD(COL-1)
  END
PBT=GPROF-OVHD
END
TOTAL=MAR+JUN+SEP+DEC
TAX=TAXRAT*PBT(TOTAL)
PAT=PBT(TOTAL)-TAX
EARN=PAT/12000

R;
T CONTROL2 EMPCON

DATA
OPEN DATA2
READ TUNITS AS CHEM46
CLOSE
MKTSHR(MAR) .025,.075,.15,.25
UCOST 6.3
PRICE 10
END
FIT
SIMPLE REGRESSION
TUNITS
COL
QTR1,QTR20
BEST
END
4
YES
TUNITS
MAR
END
RUN MODEL2
PRINT FROM REPORT2

R;
T DATA2 EMPDATA

CHEM46 32450 33270 32990 33440 &
33560 35250 35360 35500 &
35170 36660 36540 36770 &
37280 36820 37410 38580 &
38160 38220 38830 39430

R;
T REPORT2 EMPREP
```

```
SELECT MAR:TOTAL
COLUMNWIDTH 10
TITLE 1 CENTER "CHEMICAL DIVISION"
TITLE 2 CENTER "INCOME STATEMENT"
TITLE 3 CENTER /"PRODUCT 4792"//
POSITION 5
PRINT UNITS
SKIP
PREFIX "$"
PRINT SALES
PREFIX
PRINT COS
LINE
PRINT /,GPROF,OVHD
LINE
PRINT /,PBT,TAX@5,/,PAT@5,/
DECIMAL 2
PRINT EARN@5

R;
T MODEL5 EMPMOD

OPTION SECTION
MOD00010
CONTROL CONTROL5
COLUMN SECTION
CREATE QTR 1:20 "HISTORY"
MAR     "PERIOD/ENDING/3-31"
JUN     "PERIOD/ENDING/6-30"
SEP     "PERIOD/ENDING/9-30"
DEC     "PERIOD/ENDING/12-31"
TOTAL  "TOTAL/FOR/YEAR"
ROW SECTION
MKTSHR INPUT "MARKET SHARE"
TUNITS     "TOTAL MARKET"
UNITS      "UNITS SOLD"
SALES      "SALES REVENUE"
COS        "COST OF SALES"
GPROF      "GROSS PROFIT"
OVHD       "OVERHEAD EXPENSE" 10000*
PBT        "PROFIT BEFORE TAX"
SCALAR SECTION
PRICE INPUT "UNIT COST"
UCOST INPUT "UNIT COST"
TAXRAT "TAX RATE" .48
TAX        "INCOME TAX"
PAT        "PROFIT AFTER TAX"
EARN       "EARNINGS PER SHARE"
RULES SECTION
FOR COL=MAR TO DEC DO
   UNITS=ROUND(MKTSHR(COL)*TUNITS(COL),0)
   SALES=UNITS*PRICE
   COS=UNITS*UCOST
   GPROF=SALES-COS
   IF COL NE MAR THEN DO
       IF SALES GT SALES(COL-1) THEN &
          OVHD=OVHD(COL-1)+.05*(SALES-SALES(COL-1))
       ELSE OVHD=OVHD(COL-1)
  END
   PBT=GPROF-OVHD
END
TOTAL=MAR+JUN+SEP+DEC
TAX=TAXRAT*PBT(TOTAL)
PAT=PBT(TOTAL)-TAX
EARN=PAT/12000

R;
T CONTROL5 EMPCON

DATA
OPEN DATA5
READ TUNITS AS NEWLUB
CLOSE
MKTSHR(MAR) .10,.11,.15,.25
UCOST 5.2
PRICE 11
END
FIT
SIMPLE REGRESSION
TUNITS
COL
QTR1,QTR20
BEST
END
4
YES
TUNITS
MAR
END
RUN MODEL5

R;
T DATA5 EMPDATA

NEWLUB 40000 42400 43200 43700 &
44700 45200 46100 46300 &
47000 47200 47500 48000 &
49200 49500 51000 52000 &
53000 53200 54000 54500

R;
T REPORT5 EMPREP

SELECT MAR:TOTAL
COLUMNWIDTH 10
TITLE 1 CENTER "CHEMICAL DIVISION"
TITLE 2 CENTER "PROFORMA INCOME STATEMENT"
TITLE 3 CENTER /"PRODUCT 4000"//
POSITION 5
PRINT UNITS
SKIP
PREFIX "$"
PRINT SALES
PREFIX
PRINT COS
LINE
PRINT /,GPROF,OVHD
LINE
PRINT /,PBT,TAX@5,/,PAT@5,/
DECIMAL 2
PRINT EARN@5

R;
```

EMPIRE HANDOUT TWO

THIS IS THE ASSIGNMENT USING THE EMPIRE SYSTEM. YOU ARE GOING TO HAVE ALL THE FILES NECESSARY AVAILABLE TO YOU IN YOUR CMS VIRTUAL MACHINE. WHAT YOU ESSENTIALLY HAVE TO DO IS FOLLOW THE INSTRUCTIONS THAT APPEAR IN THIS TERMINAL SESSION.

TO SIGN-ON TO THE COMPUTER, YOU FOLLOW THE SAME PROCEDURE USED FOR YOUR SCRIPT ASSIGNMENT:

```
L VMXXX  (WHERE XXX IS THE LAST THREE DIGITS (OR LETTERS) OF
              YOUR ACCOUNT NUMBER).
             PRESS RETURN KEY
ENTER ACCOUNT NUMBER
              PRESS RETURN KEY
ENTER PASSWORD
               PRESS RETURN KEY
ENTER SS NUMBER
                PRESS RETURN KEY
```

THEN FOLLOW THESE SESSION COMMANDS:

```
SPOOL CONSOLE TO * START
R;
restor lm
CONNECT= 00:00:33 TOTCPU= 000:01.28
CONNECT COST $.01  CPU COST $.26  I/O COST $.18
TOTAL COST $.45  ACCOUNT BALANCE $503.68
STORAGE = 01024K
VM/SP REL 3 01/09/84 03:46
CP TERM ESCAPE \
CP TERM CHARDEL
R;
set ldrtbls 5
R;
empire
FILE 'EMPIRE SYNONYM Y' NOT FIXED,80 CHAR RECORDS.

 -----------
 E M P I R E
  TRANSLATOR
   VER 3B
 (C)1982 ADR
 -----------


 TRANSLATOR>
exe finansum
```

```
 ***TRANSLATION COMPLETED***


R;

***LOADING FINANSUM, PLEASE WAIT...


 -----------
 E M P I R E
  EXECUTIVE
    VER 3B
 (C)1982 ADR
 -----------



 <PROCESSING CONTROL FILE>

 EXEC>
DATA
 DATA>
OPEN FSUMDATA
 DATA>
READ NSALES
 DATA>
READ ECOSTS
 DATA>
READ MATSER
 DATA>
READ DEP
 DATA>
READ TAXES
 DATA>
READ INTINC
 DATA>
READ INTEXP
 DATA>
READ CLCOST
 DATA>
READ FLDEXP
 DATA>
READ INCTAX
 DATA>
CLOSE
 DATA>
END
 <CURRENT DATA "SAVED" INTO BACKUP WORKSPACE>
 EXEC>
RUN FINANSUM

 EXEC>
PRINT FROM REP
 ADJUST PAPER, THEN ENTER A CARRIAGE RETURN TO PROCEED=>
```

```
(PLEASE OO ENTER CARRIAGE RETURN)


 JUSTIN ALLEN PETROLEUM CORP AND SUBSIDIARY COMPANIES
 TEN YEAR FINANCIAL SUMMARY

                                                                   SUBTOTAL
FOR  SUBT
                                                                  1970-1974
197
                                                                 -----------
-- -----
 (DOLLARS IN MILLIONS)
 EARNINGS STATISTICS
 NET SALES.................................
18,536.7
 COSTS AND EXPENSES:
 OPERATING CHARGES:
 EMPLOYMENT COSTS..........................
7,965.2
 MATERIALS AND SERVICE.....................
7,829.1
 OEPRECIATION..............................
912.1
  TAXES, OTHER THAN EMPLOYMENT AND
 INCOME TAXES..............................
297.7
                                                                 ------  ---
-- -----
                                                                    $
17,004.1    $
                                                                 -----------
-- -----
 OPERATING INCOME..........................                         $
1,532.6     $
 OTHER INCOME(EXPENSE)
 INTEREST, OIVIDENDS AND OTHER INCOME......
196.9
 INTEREST AND OTHER OEBT CHARGES...........                              (
199.0)
 ESTIMATED CLOSEDOWN COST..................                               -
 FLOOD EXPENSE.............................                               -
                                                                 -----------
-- -----
 INCOME(LOSS) BEFORE INCOME TAXES..........                         $
1,530.5
 INCOME TAXES..............................
617.0
                                                                 -----------
-- -----
 NET INCOME (LOSS)
 AMOUNT....................................                              $
913.5
 EXEC>

 <CONTROL FILE COMPLETED>

 EXEC>
exe model2
R;
FILE 'EMPIRE SYNONYM Y' NOT FIXED,80 CHAR RECORDS.

 -----------
 E M P I R E
  TRANSLATOR
   VER 3B
 (C)1982 ADR
 -----------


 TRANSLATOR>
EXE MODEL2


 ***TRANSLATION COMPLETED***


R;

***LOADING MODEL2, PLEASE WAIT...


 -----------
 E M P I R E
  EXECUTIVE
    VER 3B
 (C)1982 ADR
 -----------


 <PROCESSING CONTROL FILE>

 EXEC>
DATA
 DATA>
OPEN DATA2
 DATA>
READ TUNITS AS CHEM46
$
 DATA>
CLOSE
 DATA>
MKTSHR(MAR) .025,.075,.15,.25
 DATA>
```

```
UCOST 6.3
 DATA>
PRICE 10
 DATA>
END
 <CURRENT DATA "SAVED" INTO BACKUP WORKSPACE>
 EXEC>
FIT

 ANALYSIS OPTION?
SIMPLE REGRESSION
 DEPENDENT ITEM NAME?
TUNITS
 INDEPENDENT ITEM NAME, OPTIONAL OFFSET?
COL
 ENTER FIRST & LAST COLUMNS TO BE USED:
QTR1,QTR20

 CURVE TYPE?
BEST
 CURVE TYPE                                        R**2        A-COEFF
B-COEFF
  1  TUNITS=A+B*COL                                .958      32410.207
349.932
  2  TUNITS=A*EXP (B*COL)                          .953      32516.371
.010
  3  TUNITS=A*(COL**B)                             .888      31157.781
.069
  4  TUNITS=A+B/COL                                .533      37327.836
6911.773
  5  TUNITS=1/(A+B*COL)                            .947           .000
-.000
  6  TUNITS=COL/(A+B*COL)                          .568           .000
.000

 LOWEST MSE ACHIEVED WITH METHOD=  1
         TUNITS=32410.207+349.932*COL

 MEAN SQUARED ERROR= 180153.938
 MEAN PERCENT ERROR= -.015
 MEAN ABSOLUTE PERCENT ERROR= 1.061
 DURBIN-WATSON= 1.886


 CURVE TYPE?
END
 NUMBER OF COLUMNS TO FORECAST?
4
 STORE THE FORECAST?
YES
 NAME OF ITEM FOR STORING VALUES?
TUNITS
 STARTING COLUMN FOR STORING VALUES?
MAR
```

```
 COLUMN          VALUE

     21        39758.781
     22        40108.711
     23        40458.645
     24        40808.578

 <CURRENT DATA "SAVED" INTO BACKUP WORKSPACE>

 ANALYSIS OPTION?
END
 EXEC>
RUN MODEL2

 EXEC>
PRINT FROM REPORT2
 ADJUST PAPER, THEN ENTER A CARRIAGE RETURN TO PROCEED=>
(PLEASE DO ENTER CARRIAGE RETURN)
 MODEL:MODEL2     DATE: 03/15/84     12:06                       PAGE: 1

                             CHEMICAL DIVISION
                              INCOME STATEMENT

                               PRODUCT 4792


   PERIOD    PERIOD    PERIOD    PERIOD                                TOTAL
   ENDING    ENDING    ENDING    ENDING                                  FOR
    3-31      6-30      9-30     12-31                                  YEAR
 --------- --------- --------- ---------                          ---------
       994     3,008     6,069    10,202  UNITS SOLD                  20,273

  $ 9,940  $ 30,080  $ 60,690 $ 102,020  SALES REVENUE            $ 202,730
    6,262    18,950    38,235    64,273  COST OF SALES              127,720
 --------- --------- --------- ---------                          ---------

    3,678    11,130    22,455    37,747  GROSS PROFIT                75,010
    5,000     6,007     7,538     9,604  OVERHEAD EXPENSE            28,149
 --------- --------- --------- ---------                          ---------

  ( 1,322)    5,123    14,918    28,143  PROFIT BEFORE TAX           46,862
                                         INCOME TAX                  22,494

                                         PROFIT AFTER TAX            24,368

                                         EARNINGS PER SHARE            2.03

 EXEC>

 <CONTROL FILE COMPLETED>

 EXEC>
```

```
simulate

 ENTER ITEMS TO BE SIMULATED (ONE PER LINE)
  TYPE "END" WHEN FINISHED

 SUPPLY ITEM NAME (OPTIONALLY WITH SUBSCRIPT), AND DISTRIBUTION TYPE,
 FOLLOWED BY THE KEYWORD "DISCRETE", IF DISCRETE POINTS ARE DESIRED.
 NOTE: IF THE SUBSCRIPT IS OMITTED, ALL ELEMENTS ARE SIMULATED.

 SIM>
mktshr(mar),uniform
 ENTER MINIMUM AND MAXIMUM VALUES:
.025,.25
 SIM>
end

 ENTER ITEMS TO BE OBSERVED (ONE PER LINE)
 TYPE "END" WHEN FINISHED

 SUPPLY ITEM NAME (OPTIONALLY WITH SUBSCRIPT).
 NOTE: IF THE SUBSCRIPT IS OMITTED, ALL ELEMENTS ARE OBSERVED.

 SIM>
earn
 SIM>
end

 ENTER NUMBER OF RANDOM TRIALS:
100
 PRINT TABLE OF INTERMEDIATE RESULTS (YES OR NO)?
no


 ITEM(S/S)                 MIN.            MAX.            MEAN
STD.
                           VALUE           VALUE           VALUE
DEV.

 INPUT ITEMS SIMULATED:

 MKTSHR(MAR)               .026            .250            .139
.063

 COMPUTED ITEMS OBSERVED:

 EARN                      2.041           3.593           2.872
.423


                        ***PROBABILITY OF RESULTS LESS THAN INDICATED
VALUE***

 ITEM(S/S)                 0%              25%             50%
75%
```

```
 MKTSHR(MAR)                    .026             .089             .147
.196
 EARN                          2.041            2.569            2.941
3.252

     ***CORRELATIONS OF "COMPUTED" VS "SIMULATED" ITEMS***

                                EARN

 MKTSHR(MAR)                    .999

 EXEC>
what is earn if price = 11

 EARN                  2.874


 EXEC>

what is earn if ucost = 6

 EARN                  2.294


 EXEC>
what is earn if price = 11 and ucost = 6

 EARN                  3.137


 EXEC>
what is earn if price = 10.5 and ucost = 6.1

 EARN                  2.628


 EXEC>
what is earn if price = 10.5 and ucost = 6

 EARN                  2.716


 EXEC>
what is earn if price = 10.6 and ucost = 6

 EARN                  2.800
 EXEC>
what impacts tax

 INITIAL VALUE OF TAX = 22493.574
 FOR EACH 1% CHANGE IN THE FOLLOWING
 THE NOTED CHANGE IN TAX WILL OCCUR:
```

```
 ITEM(S/S)                  NEW           CHANGE          PERCENT

 MKTSHR(MAR)             22518.535        24.961            .111
 MKTSHR(JUN)             22539.652        46.078            .205
 MKTSHR(SEP)             22585.734        92.160            .410
 MKTSHR(DEC)             22650.246       156.672            .697
 TUNITS(MAR)             22518.535        24.961            .111
 TUNITS(JUN)             22539.652        46.078            .205
 TUNITS(SEP)             22585.734        92.160            .410
 TUNITS(DEC)             22650.246       156.672            .697
$
 OVHD(MAR)               22397.574       -96.000           -.427
 PRICE                   23427.566       933.992           4.152
 UCOST                   21880.512      -613.063          -2.726
 TAXRAT                  22718.516       224.941           1.000


 EXEC>
what is impacted by taxrat

 FOR EACH 5.% CHANGE IN TAXRAT
 THE FOLLOWING IMPACTS WILL RESULT:
 ITEM(S/S)                  OLD             NEW            CHANGE
PERCENT

 TAX                     22493.574       23618.254       1124.680
5.000
 PAT                     24368.047       23243.367      -1124.680
-4.615
 EARN                        2.031           1.937          -.094
-4.615



 EXEC>
simulate

 ENTER ITEMS TO BE SIMULATED (ONE PER LINE)
  TYPE "END" WHEN FINISHED

 SUPPLY ITEM NAME (OPTIONALLY WITH SUBSCRIPT), AND DISTRIBUTION TYPE,
 FOLLOWED BY THE KEYWORD "DISCRETE", IF DISCRETE POINTS ARE DESIRED.
 NOTE: IF THE SUBSCRIPT IS OMITTED, ALL ELEMENTS ARE SIMULATED.

 SIM>
ucost,uniform
 ENTER MINIMUM AND MAXIMUM VALUES:
5,6.3
 SIM>
end

 ENTER ITEMS TO BE OBSERVED (ONE PER LINE)
 TYPE "END" WHEN FINISHED
 SUPPLY ITEM NAME (OPTIONALLY WITH SUBSCRIPT).
 NOTE: IF THE SUBSCRIPT IS OMITTED, ALL ELEMENTS ARE OBSERVED.

 SIM>
earn
 SIM>
end

 ENTER NUMBER OF RANDOM TRIALS:
50
 PRINT TABLE OF INTERMEDIATE RESULTS (YES OR NO)?
no

 ITEM(S/S)                MIN.           MAX.           MEAN
STD.
                          VALUE          VALUE          VALUE
DEV.

 INPUT ITEMS SIMULATED:

 UCOST                    5.029          6.271          5.659
.367

 COMPUTED ITEMS OBSERVED:

 EARN                     2.056          3.147          2.594
.322


                    ***PROBABILITY OF RESULTS LESS THAN INDICATED
VALUE***

 ITEM(S/S)                  0%             25%            50%
75%

 UCOST                    5.029          5.389          5.613
6.035
 EARN                     2.056          2.351          2.678
2.864

     ***CORRELATIONS OF "COMPUTED" VS "SIMULATED" ITEMS***

                          EARN

 UCOST                    -.999

 EXEC>
simulate

 ENTER ITEMS TO BE SIMULATED (ONE PER LINE)
  TYPE "END" WHEN FINISHED
 SUPPLY ITEM NAME (OPTIONALLY WITH SUBSCRIPT), AND DISTRIBUTION TYPE,
 FOLLOWED BY THE KEYWORD "DISCRETE", IF DISCRETE POINTS ARE DESIRED.
 NOTE: IF THE SUBSCRIPT IS OMITTED, ALL ELEMENTS ARE SIMULATED.

 SIM>
ucost, uniform
 ENTER MINIMUM AND MAXIMUM VALUES:
5,6.3
 SIM>
price,uniform
 ENTER MINIMUM AND MAXIMUM VALUES:
10,11
 SIM>
end

 ENTER ITEMS TO BE OBSERVED (ONE PER LINE)
 TYPE "END" WHEN FINISHED

 SUPPLY ITEM NAME (OPTIONALLY WITH SUBSCRIPT).
 NOTE: IF THE SUBSCRIPT IS OMITTED, ALL ELEMENTS ARE OBSERVED.

 SIM>
earn
 SIM>
end

 ENTER NUMBER OF RANDOM TRIALS:
100
 PRINT TABLE OF INTERMEDIATE RESULTS (YES OR NO)?
no


 ITEM(S/S)                 MIN.             MAX.             MEAN
STD.
                           VALUE            VALUE            VALUE
DEV.

 INPUT ITEMS SIMULATED:

 UCOST                     5.003            6.296            5.682
.394
 PRICE                    10.028           10.999           10.533
.284

 COMPUTED ITEMS OBSERVED:

 EARN                      2.143            3.863            3.023
.437


                        ***PROBABILITY OF RESULTS LESS THAN INDICATED
VALUE***
```

```
 ITEM(S/S)                    0%             25%             50%
75%

 UCOST                     5.003           5.339           5.753
6.076
 PRICE                    10.028          10.290          10.571
10.775
 EARN                      2.143           2.694           3.003
3.416

     ***CORRELATIONS OF "COMPUTED" VS "SIMULATED" ITEMS***

                           EARN

 UCOST                    -.836
 PRICE                     .617

 EXEC>
plot
 ENTER TERMINAL TYPE>
tty
 PLOT>
select mar:dec
 PLOT>
item tunits line x
 PLOT>
item tunits text
 PLOT>
go pause
 ADJUST PAPER, THEN ENTER A CARRIAGE RETURN TO PROCEED=>


        X  TOTAL MARKET

 41000. +-----------+-----------+-----------+-----------+
        :                                               :
        :                                               :
        :                                               :
        :                                         X     :
 40750. +                                               +
        :                                               :
        :                                               :
        :                                               :
        :                                               :
 40500. +                                               +
        :                             X                 :
        :                                               :
        :                                               :
        :                                               :
 40250. +                                               +
        :                                               :
        :                                               :
        :                 X                             :
```

```
        :                                               :
 40000. +                                               +
        :                                               :
        :                                               :
        :                                               :
        :                                               :
 39750. +-----X-----+-----------+-----------+-----------+
             MAR         JUN         SEP         DEC

 TUNITS     39759.      40109.      40459.      40809.

 PLOT>
new
 PLOT>
item tunits line $
 PLOT>
vs sales
 PLOT>
go pause
 ADJUST PAPER, THEN ENTER A CARRIAGE RETURN TO PROCEED=>

        $  TOTAL MARKET

 41000. +---------------+---------------+---------------+
        :                                               :
        :                                               :
        :                                               :
        :                                 $             :
 40750. +                                               +
        :                                               :
        :                                               :
        :                                               :
        :                                               :
 40500. +                                               +
        :                   $                           :
        :                                               :
        :                                               :
        :                                               :
 40250. +                                               +
        :                                               :
        :                                               :
        :         $                                     :
        :                                               :
 40000. +                                               +
        :                                               :
        :                                               :
        :                                               :
        :                                               :
 39750. +--$------------+---------------+---------------+
       0.            50000.         100000.         150000.

                          SALES REVENUE
 PLOT>
new
 PLOT>
item units line x
 PLOT>
item units text
 PLOT>
go pause
 ADJUST PAPER, THEN ENTER A CARRIAGE RETURN TO PROCEED=>


        X  UNITS SOLD

 12500. +-----------+-----------+-----------+-----------+
        :                                               :
        :                                               :
        :                                               :
        :                                               :
 10000. +                                         X     +
        :                                               :
        :                                               :
        :                                               :
        :                                               :
  7500. +                                               +
        :                                               :
        :                                               :
        :                             X                 :
        :                                               :
  5000. +                                               +
        :                                               :
        :                                               :
        :                                               :
        :                 X                             :
  2500. +                                               +
        :                                               :
        :                                               :
        :     X                                         :
        :                                               :
    0.  +-----------+-----------+-----------+-----------+
             MAR         JUN         SEP         DEC

 UNITS        994.       3008.       6069.      10202.

                          SALES REVENUE

 PLOT>
new
 PLOT>
item sales bar +
 PLOT>
go pause
 ADJUST PAPER, THEN ENTER A CARRIAGE RETURN TO PROCEED=>
```

```
        +  SALES REVENUE

125000. +-----------+-----------+-----------+-----------+
        :                                               :
        :                                               :
        :                                               :
        :                                    +++++++++++:
100000. +                                    ++++++++++++
        :                                    +++++++++++:
        :                                    +++++++++++:
        :                                    +++++++++++:
        :                                    +++++++++++:
 75000. +                                    ++++++++++++
        :                                    +++++++++++:
        :                        +++++++++++ +++++++++++:
        :                        +++++++++++ +++++++++++:
        :                        +++++++++++ +++++++++++:
 50000. +                        +++++++++++ ++++++++++++
        :                        +++++++++++ +++++++++++:
        :                        +++++++++++ +++++++++++:
        :                        +++++++++++ +++++++++++:
        :            +++++++++++ +++++++++++ +++++++++++:
 25000. +            +++++++++++ +++++++++++ ++++++++++++
        :            +++++++++++ +++++++++++ +++++++++++:
        :            +++++++++++ +++++++++++ +++++++++++:
        :+++++++++++ +++++++++++ +++++++++++ +++++++++++:
        :+++++++++++ +++++++++++ +++++++++++ +++++++++++:
    0.  +-----------+-----------+-----------+-----------+
            MAR         JUN         SEP         DEC

                          SALES REVENUE
```

```
 PLOT>
end
EXEC>
EXE MODEL5
R;
 TRANSLATOR>
exe model5

 TRANSLATING: MODEL5

      *MODEL *

 WARNING: 1 LINE(S) HAD NO LINE NUMBERS

 >LINE NUMBER(S) SET TO -1<


 ***TRANSLATION COMPLETED***


R;
```

```
***CREATING MODEL5, PLEASE WAIT...

***LOADING MODEL5, PLEASE WAIT...


 -----------
 E M P I R E
  EXECUTIVE
    VER 3B
 (C)1982 ADR
 -----------


 <PROCESSING CONTROL FILE>

 EXEC>
DATA
 DATA>
OPEN DATA5
 DATA>
READ TUNITS AS NEWLUB
 DATA>
CLOSE
 DATA>
MKTSHR(MAR) .10,.11,.15,.25
 DATA>
UCOST 5.2
 DATA>
PRICE 11
 DATA>
END
 <CURRENT DATA "SAVED" INTO BACKUP WORKSPACE>
 EXEC>
FIT

 ANALYSIS OPTION?
SIMPLE REGRESSION
 DEPENDENT ITEM NAME?
TUNITS
 INDEPENDENT ITEM NAME, OPTIONAL OFFSET?
COL
 ENTER FIRST & LAST COLUMNS TO BE USED:
QTR1,QTR20

 CURVE TYPE?
BEST
 CURVE TYPE                                          R**2         A-COEFF
B-COEFF
  1  TUNITS=A+B*COL                                  .981       40640.000
690.000
  2  TUNITS=A*EXP (B*COL)                            .978       40990.922
.014
```

```
  3  TUNITS=A*(COL**B)                              .903        38516.898
.101
  4  TUNITS=A+B/COL                                 .555        50355.930      -
13736.031
  5  TUNITS=1/(A+B*COL)                             .970             .000
-.000
  6  TUNITS=COL/(A+B*COL)                           .639             .000
.000

 LOWEST MSE ACHIEVED WITH METHOD=  1
         TUNITS=40640.000+690.000*COL

 MEAN SQUARED ERROR= 305950.000
 MEAN PERCENT ERROR= -.014
 MEAN ABSOLUTE PERCENT ERROR= .980
 DURBIN-WATSON= .877


 CURVE TYPE?
END
 NUMBER OF COLUMNS TO FORECAST?
4
 STORE THE FORECAST?
YES
 NAME OF ITEM FOR STORING VALUES?
TUNITS
 STARTING COLUMN FOR STORING VALUES?
MAR

 COLUMN           VALUE

     21        55130.000
     22        55820.000
     23        56510.000
     24        57200.000

 <CURRENT DATA "SAVED" INTO BACKUP WORKSPACE>

 ANALYSIS OPTION?
END
 EXEC>
RUN MODEL5

 EXEC>

 <CONTROL FILE COMPLETED>

 EXEC>
print from report5
 ADJUST PAPER, THEN ENTER A CARRIAGE RETURN TO PROCEED=>
 MODEL:MODEL5      DATE: 03/15/84     12:32                              PAGE: 1

                                   CHEMICAL DIVISION
                               PROFORMA INCOME STATEMENT
```

```
                              PRODUCT 4000

  PERIOD    PERIOD    PERIOD    PERIOD                                TOTAL
  ENDING    ENDING    ENDING    ENDING                                  FOR
   3-31      6-30      9-30     12-31                                  YEAR
---------  ---------  ---------  ---------                          ---------
   5,513     6,140     8,477    14,300   UNITS SOLD                  34,430

$ 60,643  $ 67,540  $ 93,247 $ 157,300   SALES REVENUE            $ 378,730
  28,668    31,928    44,080    74,360   COST OF SALES              179,036
---------  ---------  ---------  ---------                          ---------

  31,975    35,612    49,167    82,940   GROSS PROFIT               199,694
  10,000    10,345    11,630    14,833   OVERHEAD EXPENSE            46,808
---------  ---------  ---------  ---------                          ---------

  21,975    25,267    37,536    68,107   PROFIT BEFORE TAX          152,886
                                         INCOME TAX                  73,385

                                         PROFIT AFTER TAX            79,501

                                         EARNINGS PER SHARE            6.63


 EXEC>
exit

 ***E M P I R E  -  END OF SESSION***


R;
spool console close stop
CON FILE 6242 TO VEMXXX
R:

READ EMP EMP
RECORD LENGTH IS '132' BYTES.
R;

OSPRINT EMP EMP
VMXXXXXX
R;
```

ADDITIONAL EMPIRE ASSIGNMENT USING BREAKEVEN ANALYSIS

```
MODEL1 EMPMOD

OPTION SECTION
ROWTITLE 16
COLUMN 12
WIDTH 132
COLUMN SECTION
QUANT1 "QUANTITY 1"
QUANT2 "QUANTITY 2"
QUANT3 "QUANTITY 3"
QUANT4 "QUANTITY 4"
QUANT5 "QUANTITY 5"
QUANT6 "QUANTITY 6"
QUANT7 "QUANTITY 7"
QUANT8 "QUANTITY 8"
QUANT9 "QUANTITY 9"
ROW SECTION
UNITS  "         UNITS"
SALESR "SALES REVENUE"
FIXCOS "      FIX COST"
VARCOS "VARIABLE COST"
TOTCOS "    TOTAL COST"
PROFIT "PROFIT (LOSS)"
RULES SECTION
UNITS(QUANT1)=1000
FOR COL=QUANT2 TO QUANT9 DO
   UNITS(COL)=UNITS(COL-1)+1000
END
FOR COL=QUANT1 TO QUANT9 DO
   SALESR(COL)=12*UNITS(COL)
   FIXCOS(COL)=20000
   VARCOS(COL)=8*UNITS(COL)
   TOTCOS(COL)=FIXCOS(COL)+VARCOS(COL)
   PROFIT(COL)=SALESR(COL)-TOTCOS(COL)
END
```

REPORT FILE FOR ADDITIONAL EMPIRE ASSIGNMENT

REPORT1 EMPREP

```
TITLE 1 LEFT "BREAK-EVEN ANALYSIS"//
SELECT QUANT1:QUANT9
SKIP 1
PRINT UNITS
SKIP 1
PREFIX "$"
PRINT SALESR
SKIP 1
PRINT FIXCOS
SKIP 1
PRINT VARCOS
SKIP 1
PRINT TOTCOS=
SKIP 1
PRINT PROFIT
```

INSTRUCTIONS FOR USING THE IDMS DATABASE SYSTEM

As a manager for the Justin Allen Petroleum Company, you have access to the following database records. There are four types of records; product, market, market estimation, and market history records contained in your portion of the corporate database :

PRODUCT	MARKET	MKTEST	MKTHIST
FASTGAS	TEXAS NEW YORK OKLAHOMA		
CLEANGAS	MIDWEST		
SPECIALOIL			
QUIKLUB	NORTHEAST		
CHEM46	UPSTATE	1984	79THRU83
SPECIALLUB	SOUTHWEST		
POWOIL	WESTCOAST	1984	79THRU83
NOKNOCK	SOUTHEAST		
CHEAPGAS	BACKEAST MEXICO		

When you do your assignment using the database, you may not duplicate any product or market record names when you add a new record. For instance, if you wish to add a new product, do not name it NOKNOCK; make up a new name. Similarly, you may not add any BACKEAST MARKET records.

Each of the above types of records for products, markets, market estimates, and market histories contains several fields of information. For example, the POWOIL PRODUCT record contains the following fields of information:

RECORD TYPE : PRODUCT

PRODIDNO	1000
PRODNAME	POWOIL
PRODSTATUS	03

Therefore, the product's identification number is 1000, the product's name is POWOIL, and the product's status is 03, which means it is a product in full production. The valid status codes for products include 01--research, 02--trial, 03--full production, and 04--special order.

Similarly, the WESTCOAST MARKET record for POWOIL contains the following information:

```
 RECORD TYPE : MARKET
=============

 MKTIDNO      3000
 MKTNAME      WESTCOAST
 OVHDCOST     15000
 OVHDRATE     .05
 TAXRATE      .45
 SHARES        1500
 UCOST         5.00
 PRICE         8.00
```

Any market record must belong to a product record; we cannot have a market record which does not belong to a product.

The 1984 MKTEST record for POWOIL in the WESTCOAST market contains the following information:

```
 RECORD TYPE : MKTEST
=============

 FDATE       1984
 MKTSHR     .15.20.22.25
 UNITS       20000
 SALES       160000
 COS         100000
 GPROF       60000
 OVHDEXPS    20000
 PBT         40000
 TAX         18000
 PAT         22000
 EARN        2.30
```

A MKTEST (market estimate) record must belong to a particular market record; an estimate of the market cannot be done unless there is a market to be estimated.

Similarly, a MKTHIST record must belong to a particular MARKET record. The 79THRU83 MKTHIST record for POWOIL in the WESTCOAST market contains the following information:

```
 RECORD TYPE : MKTHIST
=============

 PERIOD     79THRU83
 YEAR1      12345 12543 12534 15432
 YEAR2      22222 33333 44444 55555
 YEAR3      11111 22222 33333 33333
 YEAR4      11111 22222 33333 33333
 YEAR5      11111 22222 33333 44444
```

The four figures listed in each YEAR field contain quarterly sales data for the year.

Notice that the **boldfaced** entries constitute the record's identification (RID). You use this (RID) to access the information contained in a record.

Now that you are becoming familiar with the types of data to be found in the database, you are ready to examine, add, delete, and change information in the database. First you must log onto a terminal. You will receive the prompt

```
R;
```

and you should type

RESTOR 1M

and hit the RETURN key. You will receive checkpointing information and will again receive the

```
R;
```

prompt. You should then type

DBENTER

and hit the RETURN key. You will receive a screen of information; wait until you receive the

```
     IF YOU ARE USING A COURIER TERMINAL PLEASE
     PRESS PA 2 THEN ENTER.  OTHERWISE PRESS RETURN
```

message and follow these directions. You will then receive a menu telling you the commands for using the database:

```
YOU ARE NOW RUNNING UNDER IDMS.  BELOW YOU WILL SEE DISPLAYED A
MENU OF THE AVAILABLE DB MANIPULATION COMMANDS .  BEFORE USING
ANY OF THE COMMANDS IN THE MENU IT IS RECOMMENDED THAN YOU READ A
DESCRIPTION.  THIS INFORMATION IS AVAILABLE FROM YOUR INSTRUCTOR.
( You are reading this information now )
IF YOU WANT TO LEAVE IDMS TYPE "LEAVE" THEN PRESS ENTER FOLLOWING
ANY PROMPT FOR A COMMAND.

     DBGET THE (RID) (RN) FOR (PID) IN THE (MID)

     DBDELETE THE (RID) (RN) FOR (PID) IN THE MID)

     DBCHANGE THE (RID) (RN) FOR (PID) IN THE (MID)

     DBADD A (RN) FOR (PID) IN THE (MID)


     DBMENU;      DBREPORT;      LEAVE;

TO EXECUTE A COMMAND TYPE THE COMMAND WITH THE
APPROPRIATE ARGUMENTS THEN PRESS RETURN.
```

Each of these commands is explained below. You should enter one of these commands whenever you receive the

ENTER COMMAND:

prompt.

The commands listed on the bottom row, DBMENU, DBREPORT, and LEAVE require you to simply type the command and then hit the RETURN key. Use these commands for the following purposes:

DBMENU

Use DBMENU whenever you wish to look at the menu above.

DBREPORT

DBREPORT gives you a report of all the data you have access to in the database and a listing of the changes you have made to the database during the current terminal session. You will receive the message:

THE REPORT HAS BEEN SENT TO THE OUTPUT FILE

You will not be able to look at this information until you LEAVE the database; the report will be sent to an output file called DBOUT LISTING which you can look at after you LEAVE the database. To look at the DBOUT LISTING file, you can use XEDIT, TYPE, or OSPRINT. You will recieve more on this later.

LEAVE

Use LEAVE whenever you are finished working with the database for the current session. You may then look at your DBOUT LISTING file for a record of your changes to the database.

The remaining commands enable you to work with individual records within the database. These commands require you to enter several parameters, which are explained below.

DBGET THE (RID) (RN) FOR (PID) IN THE (MID)

An example of the use of DBGET is:

DBGET THE 1984 MKTEST FOR POWOIL IN THE WESTCOAST

where

1984 is the record identification (RID).
MKTEST is the record name (RN).
POWOIL is the product identification (PID).
WESTCOAST is the market identification (MID).

You would receive the following output:

```
 RECORD TYPE : MKTEST
=============

FDATE      1984
MKTSHR     .15.20.22.25
UNITS       20000
SALES        160000
COS         100000
GPROF       60000
OVHDEXPS    20000
PBT         40000
TAX         18000
PAT         22000
EARN       2.30

THE 1984 MKTEST  RECORD
HAS BEEN SENT TO THE OUTPUT FILE
```

Another example would be:

```
DBGET THE WESTCOAST MARKET FOR POWOIL

RECORD TYPE : MARKET
=============

MKTIDNO      3000
MKTNAME      WESTCOAST
OVHDCOST     15000
OVHDRATE     .05
TAXRATE      .45
SHARES       20000
UCOST         5.00
PRICE         8.00

THE WESTCOAST MARKET  RECORD
HAS BEEN SENT TO THE OUTPUT FILE
```

As you can see from these two examples, MID was not a required parameter for the second example. An MID (market identification) was not required because we have already fully identified the market record we wished to retrieve. Also, in the second example, WESTCOAST was used as an RID; in the first example, WESTCOAST was used as an MID.

A final example would be:

```
DBGET THE POWOIL PRODUCT
```

You would receive the following information:

```
RECORD TYPE : PRODUCT
=============

PRODIDNO      1000
PRODNAME      POWOIL
PRODSTATUS    03

THE POWOIL PRODUCT  RECORD
HAS BEEN SENT TO THE OUTPUT FILE
```

In using the DBGET or any of the remaining commands, you should keep in mind the following points:

(RN) can only be one of the following: PRODUCT, MARKET, MKTHIST, or MKTEST.

(RID) identifies the particular type of RN (for example, POWOIL, WESTCOAST, 79THRU83, or 1984) you wish to retrieve.

(PID) is necessary to identify a product when you use a MARKET, MKTEST, or MKTHIST record.

(MID) is necessary to identify a market when you use a MKTEST or MKTHIST record.

DBDELETE THE (RID) (RN) FOR (PID) IN THE (MID)

DBDELETE allows you to eliminate a record (or several records) from the database. Several examples follow:

DBDELETE THE POWOIL PRODUCT

With this command you eliminate not only the POWOIL PRODUCT record, but also the WESTCOAST MARKET record for POWOIL (remember that a market cannot exist unless it is associated with a particular product; if POWOIL is eliminated, WESTCOAST does not have a product to belong to). In addition, the 79THRU83 MKTHIST record and the 1984 MKTEST record would also be eliminated since they would no longer have a WESTCOAST MARKET "owner".

DBDELETE THE WESTCOAST MARKET FOR POWOIL

This command would eliminate the WESTCOAST MARKET, 79THRU83 MKTHIST, and 1984 MKTEST records.

DBDELETE THE 1984 MKTEST FOR POWOIL IN THE WESTCOAST

This command will delete only the 1984 MKTEST record.

DBDELETE THE 79THRU83 MKTHIST FOR POWOIL IN THE WESTCOAST

This command will delete only the 79THRU83 MKTHIST record.

In using the DBDELETE command, you must be careful in deleting PRODUCT or MARKET records, since these deletions will automatically result in the deletion of records "owned" by the deleted record.

DBCHANGE THE (RID) (RN) FOR (PID) IN THE (MID)

If you wish to change a record in the database, we would recommend that you first DBGET the record so that you will be sure of the field names and their values. For instance, if you wish to change the SOUTHEAST MARKET record for the NOKNOCK product, you should first do the following:

```
DBGET THE SOUTHEAST MARKET FOR NOKNOCK
```

You will receive the following output:

```
RECORD TYPE : MARKET
=============

MKTIDNO      9000
MKTNAME      SOUTHEAST
OVHDCOST     17500
OVHDRATE     .04
TAXRATE      .52
SHARES       17500
UCOST         6.80
PRICE         9.00

THE SOUTHEAST MARKET  RECORD
HAS BEEN SENT TO THE OUTPUT FILE
```

Now you can use the DBCHANGE command to change one of the fields. If you wish to change the TAXRATE field in this record, for instance, you would type:

```
ENTER COMMAND:

DBCHANGE THE SOUTHEAST MARKET FOR NOKNOCK
```

You will be asked

```
DO YOU WANT TO CHANGE A VALUE IN MARKET RECORD?
YES/NO
```

And you may reply

```
YES
```

then you must specify the fields you wish to change and their new values:

```
TYPE THE NAME AND THE NEW VALUE FOR THE ITEM
YOU WANT TO CHANGE, THEN PRESS ENTER
(ie.) TAXRATE .48
```

If you wish to change other fields in this record, you may do so.

```
DO YOU WANT TO CHANGE ANY OTHER DATA ITEMS
IN THE MARKETRECORD? YES/NO
(here we chose not to )
NO
```

You will receive the following confirmation that the record has been changed.

```
THE SOUTHEAST MARKET RECORD
HAS BEEN CHANGED
```

DBADD A (RN) FOR (PID) IN THE (MID)

Adding a new product is a unique variation of DBADD. In order to add a new product, you must use the following format:

```
 DBADD A  NEW_PRODUCT
```

Please notice the underscore character between NEW and PRODUCT.

You will then be asked:

```
DO YOU WANT TO ADD A NEW PRODUCT RECORD?
YES/NO
YES
```

You may now add fields and values in the new record.

```
TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 PRODNAME LEADFREE
```

You will now be asked if you wish to add any other values for the different fields in the record. For a listing of the different fields which appear in the PRODUCT record type, see page 1. You do not have to add values for every field in the record; if you do not give a field a value, that field will appear with a value of blank.

```
DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE PRODUCT RECORD? YES/NO
YES

TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 PRODIDNO 34545

DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE PRODUCT RECORD? YES/NO
(we did not want to)

NO
```

```
A NEW RECORD OCCURRENCE HAS BEEN
INSERTED FOR THE PRODUCT RECORD
TYPE.
```

We will now DBGET the new product record we just added to show you that the DBADD worked.

```
 DBGET THE LEADFREE PRODUCT

RECORD TYPE :  PRODUCT
=============

PRODIDNO       34545
PRODNAME       LEADFREE
PRODSTATUS
```

Notice that the PRODSTATUS field is blank since we did not give that field a value.

Other examples of the use of the DBADD command follow.

```
 DBADD A  MKTHIST FOR CLEANGAS IN THE MIDWEST
```

You will be asked if you want to add a new record.

```
DO YOU WANT TO ADD A NEW MKTHIST RECORD?
YES/NO
YES
```

You must then type the field names and their values for the different fields in the record; see page 2 for the different types of fields in each record.

```
TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 PERIOD 75THRU78
```

You can now add values for the other fields in the record; you are not required to add a value for every field in the record. They will appear as blanks if you do not give them a value.

```
DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE MKTHIST RECORD? YES/NO
YES

TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 YEAR1 13234 78999 23212 87898

DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE MKTHIST RECORD? YES/NO
NO
```

You will receive confirmation that a new record has been added to the database.

```
A NEW RECORD OCCURRENCE HAS BEEN
CREATED FOR THE MKTHIST RECORD
TYPE.
```

Now we will get a copy of the new record we just added to see how it looks.

```
ENTER COMMAND:

 DBGET THE 75THRU78 MKTHIST FOR CLEANGAS IN THE MIDWEST

RECORD TYPE :  MKTHIST
=============

 PERIOD     75THRU78
 YEAR1      13234 78999 23212 87898
 YEAR2
 YEAR3
 YEAR4
 YEAR5

THE 75THRU78 MKTHIST  RECORD
HAS BEEN SENT TO THE OUTPUT FILE
```

Notice that the values which we did not initialize have been set to blanks.

Now we will add a MKTEST record.

```
ENTER COMMAND:
 DBADD A MKTEST FOR CLEANGAS IN THE MIDWEST

DO YOU WANT TO ADD A NEW MKTEST RECORD?
YES/NO
YES

TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 FDATE 1981

DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE MKTEST RECORD? YES/NO
YES

TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 MKTSHR .03.08.10.13

DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE MKTESTRECORD? YES/NO
YES

TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 EARN .5
```

```
DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE MKTEST RECORD? YES/NO
NO

A NEW RECORD OCCURRENCE HAS BEEN
INSERTED FOR THE MKTEST RECORD
TYPE.
Now we will get the record we just added.

 ENTER COMMAND
  DBGET THE 1981 MKTEST FOR CLEANGAS IN THE MIDWEST

RECORD TYPE :  MKTEST
=============

FDATE      1981
MKTSHR     .03.08.10.13
UNITS
SALES
COS
GPROF
OVHDEXPS
PBT
TAX
PAT
EARN       .5

THE 1981 MKTEST  RECORD
HAS BEEN SENT TO THE OUTPUT FILE

Finally, we will add a MARKET record.

 ENTER COMMAND:
  DBADD A  MARKET FOR CLEANGAS

DO YOU WANT TO ADD A NEW MARKET RECORD?
YES/NO
YES

TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 MKTIDNO 12

DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE MARKET RECORD? YES/NO
YES

TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 MKTNAME KANSAS

DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE MARKET RECORD? YES/NO
YES
```

```
TYPE THE NAME AND THE NEW VALUE FOR AN ITEM
IN THE NEW RECORD, THEN PRESS ENTER
 PRICE 15.00

DO YOU WANT TO ADD ANY OTHER DATA ITEMS
IN THE MARKET RECORD? YES/NO
NO
```

You will then receive confirmation that the record has been added to the database.

```
A NEW RECORD OCCURRENCE HAS BEEN
INSERTED FOR THE MARKET RECORD
TYPE.
```

Please remember when using DBADD that you must make up your own record names. If the record names you use are not unique, your terminal session may abort.

As a final reminder; when you are finished with your database tasks, you give the command LEAVE. Anything that you sent "to your output file" (via DBREPORT, DBGET,...etc.) will be stored in a CMS file called DBOUT LISTING . Then the mainframe will return you to CMS with the familier prompt:

```
R;
```

Now you may want to:

```
 TYPE DBOUT LISTING
```

To receive a display of the file at your terminal. Or you may want to:

```
 OSPRINT DBOUT LISTING
```

To receive a hard copy printout of your dboutput file, which can be picked up at the DISPATCH WINDOW in the basement of CARDWELL HALL. You will turn in this printout when you have done the following assignment.

You may also view segments of your output in the system editor XEDIT , but the details of that will not be discussed here.

We also want you to consider the fact that if everything goes smoothly in your database assignment, it may appear amazingly simple to retrieve, change, or add information to a database. This is the magic of a database management system !! For without a database, in order to do the same functions manually might require your secretary or staff to search multiple file cabinets, perhaps in different departments of the corporation, and then create a consolidated report of their search results.

Good luck !!!! and have fun !!!!

IDMS ASSIGNMENT INSTRUCTIONS

Your first assignment will be to change the following records to reflect the financial calculations made in Model 2 of your EMPIRE assignment. You should use DBCHANGE to change the **boldfaced** fields in the following market record. (See your INSTRUCTIONS FOR USING THE IDMS DATABASE SYSTEM for instructions for using the DBCHANGE command.)

```
RECORD TYPE : MARKET
============

MKTIDNO     00010000            ( You may find it helpful
MKTNAME     UPSTATE             ( to DBGET this record
OVHDCOST    05000               ( before you DBCHANGE it.
OVHDRATE    .05
TAXRATE     .48
SHARES       8000
UCOST        6.00     ( this is a boldfaced field
PRICE        10.6
```

In this market record, you have changed PRICE from 10.0 to 10.6 and UCOST from 6.3 to 6.0. As you may recall from your EMPIRE assignment these price and unit cost figures will enable you to increase earnings per share from 2.03 to 2.8, while remaining within reasonable limits of production and marketing constraints for the company.

Because you have changed the unit cost and price of the CHEM46 product in the UPSTATE market, these changes should be reflected in the market estimate for the UPSTATE market. In Model 2 of your EMPIRE assignment, you produced an income statement that contained the old market estimate information, when the price was 10 and the unit cost was 6.3. Therefore, you will need to change the current MKTEST record to reflect these new changes in the price and cost of producing CHEM46. The following **boldfaced** fields show you the changes you will need to make.

```
RECORD TYPE : MKTEST
============

FDATE       1984
MKTSHR      .02.07.15.25
UNITS       20273
SALES       214894
COS         121638
GPROF       93256
OVHDEXPS    28637
PBT         64618
TAX         31017
PAT         33602
EARN        2.80
```

An explanation of where this data came from follows.

We used EMPIRE to make a separate run of Model2 in order to show you the income statement produced as a result of changing the PRICE to 10.6 from 10 and of changing UCOST from 6.3 to 6.0. Note that the **boldfaced** fields in the income statement, which EMPIRE produced, are the changes we have made to the market estimate record for the CHEM46 product in the UPSTATE market.

```
 -----------
 E M P I R E
  EXECUTIVE
    VER 3B
 (C)1982 ADR
 -----------


MODEL:MODEL2       DATE: 03/23/84  10:31

                          CHEMICAL DIVISION
                          INCOME STATEMENT

                            PRODUCT 4792

 PERIOD    PERIOD     PERIOD     PERIOD                            TOTAL
 ENDING    ENDING     ENDING     ENDING                              FOR
   3-31      6-30       9-30      12-31                             YEAR
 -------  ---------  ---------  ---------                        --------
     994     3,008      6,069     10,202   UNITS SOLD              20,273
 $10,536  $ 31,885   $ 64,331  $ 108,141   SALES REVENUE         $214,894
   5,964    18,048     36,414     61,212   COST OF SALES          121,638
 -------  ---------  ---------  ---------                        --------
   4,572    13,837     27,917     46,929   GROSS PROFIT            93,256
   5,000     6,067      7,690      9,880   OVERHEAD EXPENSE        28,637
 -------  ---------  ---------  ---------                        --------

  ( 428)     7,769     20,228     37,049   PROFIT BEFORE TAX 64,618
                                           INCOME TAX              31,017

                                           PROFIT AFTER TAX  33,602

                                           EARNINGS PER SHARE  2.80


***E M P I R E  -  END OF SESSION***
```

Now we would like for you to add some new records to the database. The records you will be adding contain data from the run of Model 5 of your EMPIRE assignment. Refer to your INSTRUCTIONS FOR USING THE IDMS DATABASE SYSTEM in order to use the DBADD command to add the following records.

```
RECORD TYPE : PRODUCT
=============

PRODIDNO      4000
PRODNAME      NEWLUB
PRODSTATUS    03


RECORD TYPE : MARKET
=============

MKTIDNO       2000
MKTNAME       EASTCOAST
OVHDCOST      10000
OVHDRATE      .05
TAXRATE       .48
SHARES         12000
UCOST          5.20
PRICE          11.00


RECORD TYPE : MKTEST
=============

FDATE        1984
MKTSHR      .10.11.15.25
UNITS        34430
SALES        378730
COS          179036
GPROF        199694
OVHDEXPS     46808
PBT          152886
TAX          73385
PAT          79501
EARN         6.63

RECORD TYPE : MKTHIST
=============

PERIOD      79THRU83
YEAR1       40000 42400 43200 43700
YEAR2       44700 45200 46100 46300
YEAR3       47000 47200 47500 48000
YEAR4       44200 49500 51000 52000
YEAR5       53000 53200 54000 54500
```

For your information, we have reproduced portions of your Model 5 EMPIRE run and boldfaced the portions which contained data for the database records.

```
-----------
E M P I R E
 EXECUTIVE
   VER 38
(C)1982 ADR
-----------

<PROCESSING CONTROL FILE>

EXEC> DATA
DATA> OPEN DATA5
DATA> READ TUNITS AS NEWLUB
DATA> CLOSE
DATA> MKTSHR(MAR) .10,.11,.15,.25
DATA> UCOST 5.2
DATA> PRICE 11
DATA> END

<CURRENT DATA "SAVED" INTO BACKUP WORKSPACE>

EXEC> PRINT FROM REPORT5

ADJUST PAPER, THEN ENTER A CARRIAGE RETURN TO PROCEED=>


MODEL:MODEL5      DATE: 03/23/84    10:51              PAGE: 1

                         CHEMICAL DIVISION
                     PROFORMA INCOME STATEMENT

                           PRODUCT 4000


PERIOD    PERIOD    PERIOD    PERIOD                            TOTAL
ENDING    ENDING    ENDING    ENDING                              FOR
  3-31      6-30      9-30     12-31                             YEAR
------- --------- --------- ---------                          ------
  5,513     6,140     8,477    14,300  UNITS SOLD             34,430

$60,643  $ 67,540  $ 93,247 $ 157,300  SALES REVENUE  $ 378,730
 28,668    31,928    44,080    74,360  COST OF SALES    179,036
------- --------- --------- ---------                 ---------

 31,975    35,612    49,167    82,940  GROSS PROFIT     199,694
 10,000    10,345    11,630    14,833  OVERHEAD EXPENSE  46,808
------- --------- --------- ---------                 ---------

 21,975    25,267    37,536    68,107 PROFIT BEFORE TAX 152,886
                                      INCOME TAX         73,385

                                      PROFIT AFTER TAX   79,501

                                      EARNINGS PER SHARE   6.63
```

More information is gathered from portions of the MODEL5 EMPMOD file:

```
OPTION SECTION
CONTROL CONTROL5
COLUMN SECTION
        .
        .
        .
ROW SECTION
        .
        .
        .
OVHD      "OVERHEAD EXPENSE" 10000 ( this is the data for
        .                           ( the OVHDCOST field
        .                           ( in the market record.
        .
SCALAR SECTION
        .
        .
        .
TAXRAT "TAX RATE" .48             ( in the market record
        .
        .
        .
RULES SECTION
        .
        .
        .
   IF SALES GT SALES(COL-1) THEN &
       OVHD=OVHD(COL-1)+ .05 *(SALES-SALES(COL-1))
   ELSE OVHD=OVHD(COL-1) ( ↑ this is the OVHDRATE
  END                    (   in the market record
        .
        .
        .
EARN=PAT/12000        ( this is SHARES in the market record
```

Other information for the MKTHIST record is contained in the DATA5 EMPDATA file of the EMPIRE data files.

```
NEWLUB
40000 42400 43200 43700 &
44700 45200 46100 46300 &
47000 47200 47500 48000 &
49200 49500 51000 52000 &
53000 53200 54000 54500
```

We made up the PRODSTATUS of 03 for the PRODUCT record, the MKTIDNO 2000 and the MKTNAME EASTCOAST for the MARKET record, and the FDATE of 1984 for the MKTEST.

Once you have used DBADD to add a record to the database, you should use DBGET to make sure that the record was added correctly.

As a final task of this assignment you should DBDELETE the QUICKLUB product, and all it's associated records. And then, DBDELETE the market estimate record for the POWOIL product in the WESTCOAST market.

Required output from this assignment will be a printout of the DBOUT LISTING file. You must use the DBREPORT comand, before you LEAVE the database environment. For information on how to obtain this printout, see your INSTRUCTIONS FOR USING THE IDMS DATABASE.

CHEMDEV1 SCHMA

```
************************************************************
*        SCHEMA DESCRIPTION STATEMENTS                      *
************************************************************
*
 SCHEMA DESCRIPTION.
*
 SCHEMA NAME IS CHEMDEV1.
*
 AUTHOR.                ROBERT A. BIRCHARD
*
 DATE.                  06/24/83
*
 INSTALLATION.          KANSAS STATE UNIVERSITY
                        COMPUTER SCIENCE DEPARTMENT
 REMARKS.               THIS SCHEMA WAS DESIGNED TO SATISFY THE
                        STUDENT BEHAVIORAL LEARNING OBJECTIVES
                        OF A MANAGEMENT INFORMATION SYSTEMS
                        COURSE FOR INFORMATION SYSTEMS MAJORS
                        AT KANSAS STATE UNIVERSITY.
*************************************************************
*        FILE DESCRIPTION STATEMENTS                         *
*************************************************************
 FILE DESCRIPTION.
 FILE NAME IS IDMS-FILE1
                        ASSIGN TO SYS010
                        DEVICE TYPE IS 3350.
 FILE NAME IS JOURNAL
                        ASSIGN TO SYS009
                        DEVICE TYPE IS 2400.
************************************************************
*        AREA DESCRIPTION STATEMENTS                        *
************************************************************
 AREA DESCRIPTION.
 AREA NAME IS PRODUCT-REGION
        RANGE IS 1001 THRU 1010
        WITHIN IDMS-FILE1
        FROM 1 THRU 10.
 AREA NAME IS MARKET-REGION
        RANGE IS 1011 THRU 1020
        WITHIN IDMS-FILE1
        FROM 11 THRU 20.
 AREA NAME IS HIST-REGION
        RANGE IS 1021 THRU 1030
        WITHIN IDMS-FILE1
        FROM 21 THRU 30.
************************************************************
*        RECORD DESCRIPTION STATEMENTS                      *
************************************************************
 RECORD DESCRIPTION.
 RECORD NAME IS PRODUCT.
 RECORD ID IS 100.
 LOCATION MODE IS
        CALC USING PRODNAME
        DUPLICATES ARE NOT ALLOWED.
 WITHIN PRODUCT-REGION AREA.
       03                PRODIDNO         PIC X(8) .
                 COMMENT 'PRODUCT ID#,  THERE IS A UNIQUE ID#,
 -                       'FOR EACH PRODUCT IN THE FILE'.
       03                PRODNAME        PIC X(12) .
       03                PRODSTATUS     PIC X(2) .
                 COMMENT 'THE PRODUCT STATUS  THIS IS USED TO
 -                       'INDICATE THE MARKETING PHASE,
 -                       'RESEARCH=01,
 -                       'TRIAL=02,
 -                       'FULL PRODUCTION=03,
 -                       'CUSTOM ORDERS=04,
 -                       'DISCONTINUED=05'.
 RECORD NAME IS MARKET.
 RECORD ID IS 300.
 LOCATION MODE IS
        CALC USING MKTNAME
        DUPLICATES ARE NOT ALLOWED.
 WITHIN MARKET-REGION AREA.
       03                MKTIDNO          PIC X(8) .
                 COMMENT 'MARKET ID#,  THERE IS A UNIQUE ID#
 -                       'FOR EACH MARKETING REGION'.
       03                MKTNAME         PIC X(12).
       03                OVHDCOST        PIC X(5).
                 COMMENT 'OVERHEAD COST,  THE ESTIMATED COST
 -                       'FOR THE FIRST QUARTER OF THE
 -                       'FORECAST PERIOD'.
       03                OVHDRATE        PIC X(3).
                 COMMENT 'OVERHEAD INCREASE RATE,  THE PERCENT
 -                       'RATE AT WICH OVERHEAD COSTS ARE
 -                       'EXPECTED TO INCREASE DURING THE
 -                       'FORECAST PERIOD'.
       03                TAXRATE         PIC X(3).
       03                SHARES          PIC X(5).
                 COMMENT 'THE EFFECTIVE NUMBER OF COMMON STOCK
 -                       'SHARES TO BE USED FOR THE COMPUTATION
 -                       'OF EARNINGS PER SHARE'.
       03                UCOST           PIC X(5).
                 COMMENT 'UNIT COST,  THE ESTIMATED COST OF
 -                       'PRODUCING EACH UNIT OF PRODUCT'.
       03                PRICE           PIC X(5).
                 COMMENT 'UNIT PRICE,  THE PLANNED SELLING
 -                       'PRICE PER UNIT OF PRODUCT'.
 RECORD NAME IS MKTHIST.
 RECORD ID IS 310.
 LOCATION MODE IS
        VIA HIST SET.
 WITHIN HIST-REGION.
       03                PERIOD           PIC X(8).
                 COMMENT 'PERIOD INDICATES THE TIME PERIOD
```

```
-                          'OF THE HISTORICAL SALES DATA FOR
-                          'PRODUCTS IN THE MARKET PLACE, EXAMPLE:
-                          '78THRU83 INDICATES THE YERRS COVERED'.
       03          YEAR1                PIC X(23).
                       COMMENT 'YEAR 1 THRU YEAR5 EACH HOLD
-                          'HISTORICAL DATA BY QUARTER FOR ONE
-                          'YEAR WITHIN THE PERIOD'.
       03          YEAR2                PIC X(23).
       03          YEAR3                PIC X(23).
       03          YEAR4                PIC X(23).
       03          YEAR5                PIC X(23).
 RECORD NAME IS MKTEST.
 RECORD ID IS 320.
 LOCATION MODE IS
       VIA EST SET.
 WITHIN MARKET-REGION AREA.
       03          FDATE                PIC X(4).
                       COMMENT 'DATE OF THE FORECAST PERIOD'.
       03          MKTSHR               PIC X(12).
       03          UNITS                PIC X(6).
                       COMMENT 'UNITS SOLD, THE FORECAST NUMBER
-                          'OF UNITS OF PRODUCT SOLD'.
       03          SALES                PIC X(8).
                      COMMENT 'SALES REVENUE UNITS*PRICE  '.
       03          COS                  PIC X(7).
        COMMENT 'COST OF SALES UNITS*UCOST'.
       03          GPROF                PIC X(6).
                       COMMENT 'GROSS PROFIT,  SALES-COS'.
       03          OVHDEXPS             PIC X(6).
                       COMMENT 'OVERHEAD EXPENSE,  OVERHEAD
-                          'EXPENSES FOR EACH QUARTER OF THE
-                          'FORECAST PERIOD'.
       03          PBT                  PIC X(6).
                       COMMENT 'PROFIT BEFORE TAX,
-                              'GPROF-OVHDEXPS'.
       03          TAX                  PIC X(6).
                       COMMENT 'INCOME TAX EXPENSE,
-                              'PBT*TAXRATE'.
       03          PAT                  PIC X(6).
                       COMMENT 'PROFIT AFTER TAX.  PBT-TAX'.
       03          EARN                 PIC X(4).
 RECORD NAME IS DEVHIST.
 RECORD ID IS 200.
 LOCATION MODE IS
       CALC USING PRODEVNAME
       DUPLICATES ARE NOT ALLOWED.
 WITHIN HIST-REGION AREA.
       03          PRODEVNO             PIC X(8).
                       COMMENT 'PRODUCT DEVELOPMENT #,  IT IS
-                          'EQUIVALENT TO PRODID#'.
       03          PRODEVNAME           PIC X(12).
                       COMMENT 'PRODEVNAME IS EQUIVALENT TO
-                          'PRODNAME,  PRODUCT DEVELOPMENT NAME'.
       03          PATDOCNO             PIC X(8).
```

```
                          COMMENT 'PATENT DOCUMENT #,  THIS IS A
-                            'LIBRARY CATALOG NUMBER FOR PATENT
-                            'INFORMATION CONCERNING THE PRODUCT'.
       03         DEVDOCNO            PIC X(8).
                          COMMENT 'DEVELOPMENT DOCUMENT NUMBER,
-                            'THIS IS A LIBRARY CATALOG NUMBER FOR
-                            'DEVELOPMENT HISTORY INFORMATION
-                            'ABOUT THE PRODUCT'.
*************************************************************
*       SET DESCRIPTION STATEMENTS                          *
*************************************************************
 SET DESCRIPTION.
 SET NAME IS PROD-MKT.
 ORDER IS NEXT.
 MODE IS CHAIN.
 OWNER IS PRODUCT
        NEXT DBKEY POSITION IS 1.
 MEMBER IS MARKET
        MANDATORY AUTOMATIC
        NEXT DBKEY POSITION IS 1.
 SET NAME IS HIST.
 ORDER IS NEXT.
 MODE IS CHAIN.
 OWNER IS MARKET
        NEXT DBKEY POSITION IS 2.
 MEMBER IS MKTHIST
        MANDATORY AUTOMATIC
        NEXT DBKEY POSITION IS 1.
 SET NAME IS EST.
 ORDER IS NEXT.
 MODE IS CHAIN.
 OWNER IS MARKET
        NEXT DBKEY POSITION IS 3.
 MEMBER IS MKTEST
        OPTIONAL AUTOMATIC
        NEXT DBKEY POSITION IS 1.
```

SUBSCHEMA AND DMCL DESCRIPTION

NUPROD SUBSC

```
    ADD SUBSCHEMA NAME IS NUPROD
    OF SCHEMA NAME IS CHEMDEV1
    DMCL NAME IS DEVDMCL.

ADD AREA PRODUCT-REGION.
ADD AREA MARKET-REGION.
ADD AREA HIST-REGION.
ADD RECORD PRODUCT.
ADD RECORD MARKET.
ADD RECORD MKTHIST.
ADD RECORD MKTEST.
ADD SET PROD-MKT.
ADD SET HIST.
ADD SET EST.
GENERATE.
```

DEVDMCL DMCL

```
DEVICE-MEDIA DESCRIPTION.

DEVICE-MEDIA NAME IS DEVDMCL
                OF SCHEMA NAME CHEMDEV1.

AUTHOR.         ROBERT A. BIRCHARD
DATE.           JUNE 1983
INSTALLATION.   KANSAS STATE UNIVERSITY
             COMPUTER SCIENCE DEPARTMENT

REMARKS.   THE DMCL TABLES RESULTING FROM THESE SOURCE
           STATEMENTS WILL ALLOW RUN TIME ACCESS TO ALL
           FILES MAKING UP THE CHEMDEV1 DATA BASE.

BUFFER SECTION.

        BUFFER NAME IS DEVBUFFER
        PAGE CONTAINS 496 CHARACTERS
        BUFFER CONTAINS 8 PAGES.

AREA SECTION.

        COPY PRODUCT-REGION AREA.
        COPY MARKET-REGION AREA.
        COPY HIST-REGION AREA.
```

INITIAL VALUES FOR SUBSCHEMA

DBIN2 DECK

```
PRDDUCT00001000PDWOIL        03
MARKET 00003000WESTCDAST     15000.05.4520000 5.00 8.00
MKTHIST79THRU83                                                          PERIOO
MKTHIST12345 12543 12534 15432                                           YEAR1
MKTHIST22222 33333 44444 55555                                           YEAR2
MKTHIST11111 22222 33333 33333                                           YEAR3
MKTHIST11111 22222 33333 33333                                           YEAR4
MKTHIST11111 22222 33333 44444                                           YEAR5
MKTEST 1984.15.20.22.25 20000  160000100000  60000 20000 40000 18000
220002.30
PRDDUCT00002000CLEANGAS      02
MARKET 00004000MIDWEST       20000.04.4020000 4.50 7.00
PRDDUCT00007000NDKNDCK       02
MARKET 00009000SDUTHEAST     17500.04.5217500 6.80 9.00
PRDDUCT00008000CHEAPGAS      03
MARKET 00007000BACKEAST      22450.06.5215000 7.5012.50
MARKET 00008000MEXICD        10000.03.00 3000 3.40 6.00
PRDDUCT00009000SPECIALOIL    01
PRDDUCT00003000SPECIALLUB    04
MARKET 00005000SDUTHWEST     15000.05.4810000 6.3010.00
PRDDUCT00005000FASTGAS       03
MARKET 00011000TEXAS          8500.42.4812300  8.2   13
MARKET 00012000NEW YDRK      10100.33.4815000  8.3 13.5
MARKET 00013000DKLAHDMA       7600.25.48 7500  8.012.75
PRDDUCT00006000QUICKLUB      04
MARKET 00006000NDRTHEAST     13520.04.49 7500 5.20 7.50
PRDDUCT00015000CHEM46        03
MARKET 00010000UPSTATE       05000.05.48 8000  6.3   10
MKTHIST79THRU83                                                          PERIDD
MKTHIST32450 33270 32990 33440                                           YEAR1
MKTHIST33560 35250 35360 35500                                           YEAR2
MKTHIST35170 36660 36540 36770                                           YEAR3
MKTHIST37280 36820 37410 38580                                           YEAR4
MKTHIST38160 38220 38830 39430                                           YEAR5
MKTEST 1984.02.07.15.25 35000  202730127720  75010 28149 46862 22494
243682.03
```

FRONT-END EXEC ROUTINE

DBENTER EXEC

```
SET CMSTYPE HT
FMSET CHECKPT OFF
SET CMSTYPE RT
&CONTROL ALL
EXEC LINKIDMS
&STACK DEVDMCL
&STACK 1
&STACK 30
EXEC IDMSINIT DBASE
&TYPE ENTERING IDMS DATABASE FOR JUSTIN ALLEN PETRO. CORP.
&STACK FILEDEF SYSPRINT DISK DBOUT LISTING
&STACK FILEDEF PLIDUMP DUMMY
&STACK FILEDEF SYS010 DISK DATABASE FILE1 A (XTENT 30 BLOCK 496
&STACK FILEDEF SYSIN DISK DBIN2 DECK * (BLOCK 80 RECFM F LRECL 80
&STACK
EXEC IDMSRUN DEBQUERY
```

FRONT-END PROGRAM

```
DBQUERY : PROCEDURE OPTIONS(MAIN) ;
/*DMLIST*/
/*SCHEMA_COMMENTS*/

DECLARE
    1      CARDIN_REC              STATIC,
         2 TYP                CHAR (7),
         2 FILL1                CHAR (58),
         2 FTYP                CHAR (6),
         2 FILL2                CHAR (9),

    END_OF_AREA       CHAR (4) INIT ('0307'),
    END_OF_SET      CHAR (4) INIT ('0307'),
    INREC             CHAR (80) DEFINED CARDIN_REC,
    OK             CHAR (4) INIT ('0000');

    DECLARE (TRUE)  BIT(1) INIT ('1'B);
    DECLARE FOUND   BIT(1) INIT ('0'B) EXTERNAL;
   DECLARE DO_CMD   BIT(1)  EXTERNAL;
    DECLARE CHG_OK   BIT(1) INIT ('1'B) EXTERNAL;
    DECLARE NOGIT   BIT(1) INIT ('0'B);
    DECLARE (FALSE) BIT(1) INIT ('0'B);

    DECLARE      CMD     CHAR(72)      EXTERNAL,
         CTYP     CHAR(8)       VARYING EXTERNAL,
         RID      CHAR(12)      VARYING EXTERNAL,
         RN      CHAR(8)       VARYING EXTERNAL,
         PID      CHAR(12)      VARYING EXTERNAL,
         MID      CHAR(12)      VARYING EXTERNAL;
    DECLARE TCARD      CHAR(72)      VARYING EXTERNAL,
         TSTRING     CHAR(12)      VARYING EXTERNAL;
    DECLARE
         1    PRODBUF_REC                   STATIC,
              3      P1                   CHAR(8)  INIT(' '),
              3      P2                   CHAR(12) INIT(' '),
              3      P3                   CHAR(2)  INIT(' '),

         1    MKTBUF_REC                   STATIC,
              3      M1                   CHAR(8)  INIT(' '),
              3      M2                   CHAR(12) INIT(' '),
              3      M3                   CHAR(5)  INIT(' '),
              3      M4                   CHAR(3)  INIT(' '),
              3      M5                   CHAR(3)  INIT(' '),
              3      M6                   CHAR(5)  INIT(' '),
              3      M7                   CHAR(5)  INIT(' '),
              3      M8                   CHAR(5)  INIT(' '),

         1    MKTHISTBUF_REC                 STATIC,
              3      MH1                   CHAR(8)  INIT(' '),
              3      MH2                   CHAR(23) INIT(' '),
              3      MH3                   CHAR(23) INIT(' '),
              3      MH4                   CHAR(23) INIT(' '),
              3      MH5                   CHAR(23) INIT(' '),
              3      MH6                   CHAR(23) INIT(' '),
        1    MKTESTBUF_REC                    STATIC,
                  3       ME1                  CHAR(4)   INIT(' '),
                  3       ME2                  CHAR(12)  INIT(' '),
                  3       ME3                  CHAR(6)   INIT(' '),
                  3       ME4                  CHAR(8)   INIT(' '),
                  3       ME5                  CHAR(7)   INIT(' '),
                  3       ME6                  CHAR(6)   INIT(' '),
                  3       ME7                  CHAR(6)   INIT(' '),
                  3       ME8                  CHAR(6)   INIT(' '),
                  3       ME9                  CHAR(6)   INIT(' '),
                  3       ME10                 CHAR(6)   INIT(' '),
                  3       ME11                 CHAR(4)   INIT(' ');

 DECLARE
     PRODBUF            CHAR(22)      DEFINED      PRODBUF_REC,
     MKTBUF            CHAR(46)      DEFINED      MKTBUF_REC,
     MKTHISTBUF       CHAR(123)      DEFINED
MKTHISTBUF_REC,
     MKTESTBUF       CHAR(71)      DEFINED      MKTESTBUF_REC;

 DECLARE (IDMS,ABORT) OPTIONS (INTER,ASM) ENTRY;
 DECLARE ( NUPROD SUBSCHEMA, CHEMDEV1 SCHEMA VERSION 1)
     MODE (BATCH);
 INCLUDE IDMS ( SUBSCHEMA_DESCRIPTION);

 DECLARE
     PRODUCT_REC       CHAR (24)      DEFINED PRODUCT,
     MARKET_REC        CHAR (48)      DEFINED MARKET,
     MKTEST_REC        CHAR (71)       DEFINED MKTEST,
     MKTHIST_REC       CHAR (123)      DEFINED MKTHIST;

 INCLUDE IDMS (SUBSCHEMA_BINDS);
 SUBSCHEMA_CTRL.PROGRAM = 'DBQUERY' ;
 BIND RUN_UNIT;
 BIND RECORD (PRODUCT);
 BIND RECORD (MARKET);
 BIND RECORD (MKTEST);
 BIND RECORD (MKTHIST);

 INCLUDE IDMS(IDMS_STATUS);
 READY EXCLUSIVE UPDATE;
 CALL IDMS_STATUS;

 /* BEGIN MAIN PROGRAM */

 CALL DBUILD;
 CALL DBMENU;
 DO UNTIL (CMD = 'LEAVE     ');
     DISPLAY (' ');
     DISPLAY (' ENTER COMMAND')
          REPLY (CMD);
     DISPLAY (' ');
 CALL PROC_CMD;
```

```
END; /* DO UNTIL COM = 'LEAVE     ' */

FINISH;
CALL IDMS_STATUS;



DBCHG_REC : PROCEDURE;
    DECLARE RN CHAR(8) VARYING EXTERNAL;
         IF RN = 'PRODUCT' THEN
              DO;
                   MODIFY RECORD (PRODUCT);
                   CALL IDMS_STATUS;
              END;
         ELSE IF RN = 'MARKET' THEN
              DO;
                   MODIFY RECORD (MARKET);
                   CALL IDMS_STATUS;
              END;
         ELSE IF RN = 'MKTEST' THEN
              DO;
                   MODIFY RECORD (MKTEST);
                   CALL IDMS_STATUS;
              END;
         ELSE IF RN = 'MKTHIST' THEN
              DO;
                   MODIFY RECORD (MKTHIST);
                   CALL IDMS_STATUS;
              END;
END DBCHG_REC;



DBCHG_VAL :  PROCEDURE;

DECLARE DN     CHAR(12)       VARYING EXTERNAL,
   CHG_OK BIT(1) EXTERNAL,
    DVAL     CHAR(24)       VARYING EXTERNAL;
 CHG_OK = TRUE;

IF RN = 'PRODUCT' THEN
    DO;
              IF DN = 'PRODIDNO'    THEN    PRODIDNO     =
DVAL;
         ELSE IF DN = 'PRODNAME'    THEN    PRODNAME     =
DVAL;
         ELSE IF DN = 'PRODSTATUS' THEN    PRODSTATUS   =
DVAL;
              ELSE DO;
                   DISPLAY (DN ^^ '  IS NOT A CORRECT');
                   DISPLAY ('DATA ITEM NAME');
                   DISPLAY ('FOR THE PRODUCT RECORD');
                   CHG_OK = FALSE;
              END;
    END; /* RN = 'PRODUCT' */
ELSE IF RN = 'MARKET' THEN
    DO;
             IF DN = 'MKTIDNO'   THEN MKTIDNO  = DVAL;
        ELSE IF DN = 'MKTNAME'   THEN MKTNAME  = DVAL;
        ELSE IF DN = 'OVHDCOST'  THEN OVHDCOST = DVAL;
        ELSE IF DN = 'OVHDRATE'  THEN OVHDRATE = DVAL;
        ELSE IF DN = 'TAXRATE'   THEN TAXRATE  = DVAL;
        ELSE IF DN = 'SHARES'    THEN SHARES   = DVAL;
        ELSE IF DN = 'UCOST'     THEN UCOST    = DVAL;
        ELSE IF DN = 'PRICE'     THEN PRICE    = DVAL;
             ELSE DO;
                  DISPLAY (DN ^^ '  IS NOT A CORRECT');
                  DISPLAY ('DATA ITEM NAME');
                  DISPLAY ('FOR THE MARKET RECORD');
                  CHG_OK = FALSE;
             END;
    END; /* RN = 'MARKET' */
ELSE IF RN = 'MKTHIST' THEN
    DO;
             IF DN = 'PERIOD' THEN PERIOD   = DVAL;
        ELSE IF DN = 'YEAR1' THEN YEAR1 = DVAL;
        ELSE IF DN = 'YEAR2' THEN YEAR2 = DVAL;
        ELSE IF DN = 'YEAR3' THEN YEAR3 = DVAL;
        ELSE IF DN = 'YEAR4' THEN YEAR4 = DVAL;
        ELSE IF DN = 'YEAR5' THEN YEAR5 = DVAL;
             ELSE DO;
                  DISPLAY (DN ^^ '  IS NOT A CORRECT');
                  DISPLAY ('DATA ITEM NAME');
                  DISPLAY ('FOR THE MKTHIST RECORD');
                  CHG_OK = FALSE;
             END;
    END; /* RN = 'MKTHIST' */
ELSE IF RN = 'MKTEST' THEN
    DO;
             IF DN = 'FDATE'     THEN FDATE    = DVAL;
        ELSE IF DN = 'MKTSHR'    THEN MKTSHR   = DVAL;
        ELSE IF DN = 'UNITS'     THEN UNITS    = DVAL;
        ELSE IF DN = 'SALES'     THEN SALES    = DVAL;
        ELSE IF DN = 'COS'       THEN COS      = DVAL;
        ELSE IF DN = 'GPROF'     THEN GPROF    = DVAL;
        ELSE IF DN = 'PBT'       THEN PBT      = DVAL;
        ELSE IF DN = 'PAT'       THEN PAT      = DVAL;
        ELSE IF DN = 'OVHDEXPS'  THEN OVHDEXPS = DVAL;
        ELSE IF DN = 'TAX'       THEN TAX      = DVAL;
        ELSE IF DN = 'EARN'      THEN EARN     = DVAL;
             ELSE DO;
                  DISPLAY (DN ^^ '  IS NOT A CORRECT');
                  DISPLAY ('DATA ITEM NAME');
                  DISPLAY ('FOR THE MKTEST RECORD');
                  CHG_OK = FALSE;
             END;
    END; /* RN = 'MKTEST' */
END DBCHG_VAL; /* END OF PROCEDURE CHGVAL */
```

```
DBPRINT : PROCEDURE (PBUF,MBUF,MEBUF,MHBUF);
DECLARE      PBUF       CHAR (22),
    MBUF         CHAR (46),
    MEBUF        CHAR (71),
    MHBUF        CHAR (123);

 PUT EDIT (P1,M1,ME1,MH1)

(COL(1),A(8),X(6),A(8),X(10),A(4),X(10),A(8));
 PUT EDIT (P2,M2,ME2,MH2)

(COL(1),A(12),X(2),A(12),X(2),A(12),X(2),A(23));
 PUT EDIT (P3,M3,ME3,MH3)

(COL(1),A(2),X(12),A(5),X(9),A(6),X(6),A(23));
 PUT EDIT (M4,ME4,MH4)
(COL(15),A(3),X(11),A(8),X(7),A(23));
 PUT EDIT (M5,ME5,MH5)
(COL(15),A(3),X(11),A(6),X(8),A(23));
 PUT EDIT (M6,ME6,MH6)
(COL(15),A(5),X(9),A(6),X(8),A(23));
 PUT EDIT (M7,ME7)        (COL(15),A(5),X(9),A(6));
 PUT EDIT (M8,ME8)        (COL(15),A(5),X(9),A(6));
 PUT EDIT (ME9)             (COL(29),A(6));
 PUT EDIT (ME10)             (COL(29),A(6));
 PUT EDIT (ME11)        (COL(29),A(4));
 PUT SKIP;


 PBUF        = ' ';
 MBUF        = ' ';
 MEBUF       = ' ';
 MHBUF       = ' ';

 END DBPRINT;



 DBINIT : PROCEDURE(RN);
     DECLARE RN CHAR(8) VARYING;
    DECLARE TREC CHAR(72) INIT(' ');

 IF RN = 'PRODUCT' THEN
     DO;
          PRODUCT_REC = SUBSTR(INREC,8);
          STORE RECORD (PRODUCT);
          CALL IDMS_STATUS;
     END; /* RN = 'PRODUCT' */
 ELSE IF RN = 'MARKET' THEN
     DO;
          MARKET_REC = SUBSTR(INREC,8);
          STORE RECORD (MARKET);
          CALL IDMS_STATUS;
     END; /* RN = 'MARKET' */
 ELSE IF RN = 'MKTEST' THEN
     DO;
          MKTEST_REC = SUBSTR(INREC,8);
          STORE RECORD (MKTEST);
          CALL IDMS_STATUS;
     END; /* RN = 'MKTEST' */


 ELSE IF RN = 'MKTHIST' THEN
     DO;
          IF FTYP = 'PERIOD'      THEN PERIOD      =
SUBSTR(INREC,8);
     ELSE      IF FTYP = 'YEAR1'       THEN YEAR1      =
SUBSTR(INREC,8);
     ELSE      IF FTYP = 'YEAR2'       THEN YEAR2      =
SUBSTR(INREC,8);
     ELSE      IF FTYP = 'YEAR3'       THEN YEAR3      =
SUBSTR(INREC,8);
     ELSE      IF FTYP = 'YEAR4'       THEN YEAR4      =
SUBSTR(INREC,8);
     ELSE      IF FTYP = 'YEAR5'       THEN
          DO;
               YEAR5 = SUBSTR(INREC,8);
               STORE RECORD (MKTHIST);
               CALL IDMS_STATUS;
          END;
     END; /* RN = 'MKTHIST' */
 ELSE DISPLAY ( TYP ^^ '  IS NOT A CORRECT RECORD NAME.');
 TREC = SUBSTR(INREC,8);
 END DBINIT;



 DBDELETE : PROCEDURE(RN);
  DECLARE RN CHAR(8) VARYING;

     IF RN = 'PRODUCT' THEN
          DO;
               ERASE RECORD (PRODUCT) ALL;
               CALL IDMS_STATUS;
          END;
     ELSE IF RN = 'MARKET' THEN
          DO;
               ERASE RECORD (MARKET) ALL;
               CALL IDMS_STATUS;
          END;
     ELSE IF RN = 'MKTEST' THEN
          DO;
               ERASE RECORD (MKTEST) ALL;
               CALL IDMS_STATUS;
          END;
```

```
    ELSE IF RN = 'MKTHIST' THEN
         DO;
              ERASE RECORD (MKTHIST) ALL;
              CALL IDMS_STATUS;
         END;
END DBDELETE;



DBGET: PROCEDURE(RN);
DECLARE RN CHAR(8) VARYING;
DISPLAY ('RECORD TYPE : ' ^^ RN );
DISPLAY ('=============');
DISPLAY (' ');
PUT EDIT ('RECORD TYPE : ' ^^ RN ) (COL(1),A);
PUT EDIT ('=============') (COL(1),A);
PUT SKIP;
IF RN = 'PRODUCT' THEN DO;
     DISPLAY ('PRODIDNO  ' ^^ PRODIDNO);
     DISPLAY ('PRODNAME      ' ^^ PRODNAME);
     DISPLAY ('PRODSTATUS    ' ^^ PRODSTATUS);
     DISPLAY (' ');
     PUT EDIT ('PRODIDNO  ' ^^ PRODIDNO) (COL(1),A);
     PUT EDIT ('PRODNAME      ' ^^ PRODNAME) (COL(1),A);
     PUT EDIT ('PRODSTATUS    ' ^^ PRODSTATUS) (COL(1),A);
END; /* RN = 'PRODUCT' */
ELSE IF RN = 'MARKET' THEN DO;
     DISPLAY ('MKTIDNO  ' ^^ MKTIDNO);
     DISPLAY ('MKTNAME       ' ^^ MKTNAME);
     DISPLAY ('OVHDCOST      ' ^^ OVHDCOST);
     DISPLAY ('OVHDRATE      ' ^^ OVHDRATE);
     DISPLAY ('TAXRATE       ' ^^ TAXRATE);
     DISPLAY ('SHARES        ' ^^ SHARES);
     DISPLAY ('UCOST         ' ^^ UCOST);
     DISPLAY ('PRICE         ' ^^ PRICE);
     DISPLAY (' ');
     PUT EDIT ('MKTIDNO       ' ^^ MKTIDNO) (COL(1),A);
     PUT EDIT ('MKTNAME       ' ^^ MKTNAME) (COL(1),A);
     PUT EDIT ('OVHDCOST      ' ^^ OVHDCOST) (COL(1),A);
     PUT EDIT ('OVHDRATE      ' ^^ OVHDRATE) (COL(1),A);
     PUT EDIT ('TAXRATE       ' ^^ TAXRATE) (COL(1),A);
     PUT EDIT ('SHARES        ' ^^ SHARES) (COL(1),A);
     PUT EDIT ('UCOST         ' ^^ UCOST) (COL(1),A);
     PUT EDIT ('PRICE         ' ^^ PRICE) (COL(1),A);
END; /* RN = 'MARKET' */
ELSE IF RN = 'MKTEST' THEN DO;
     DISPLAY ('FDATE      ' ^^ FDATE);
     DISPLAY ('MKTSHR     ' ^^ MKTSHR);
     DISPLAY ('UNITS      ' ^^ UNITS);
     DISPLAY ('SALES      ' ^^ SALES);
     DISPLAY ('COS        ' ^^ COS);
     DISPLAY ('GPROF      ' ^^ GPROF);
     DISPLAY ('OVHDEXPS   ' ^^ OVHDEXPS);
     DISPLAY ('PBT        ' ^^ PBT);
     DISPLAY ('TAX         ' ^^ TAX);
     DISPLAY ('PAT         ' ^^ PAT);
     DISPLAY ('EARN        ' ^^ EARN);
    DISPLAY (' ');
    PUT EDIT ('FDATE        '  ^^ FDATE) (COL(1),A);
    PUT EDIT ('MKTSHR       '  ^^ MKTSHR) (COL(1),A);
    PUT EDIT ('UNITS        '  ^^ UNITS) (COL(1),A);
    PUT EDIT ('SALES        '  ^^ SALES) (COL(1),A);
    PUT EDIT ('COS          '  ^^ COS)   (COL(1),A);
    PUT EDIT ('GPROF        '  ^^ GPROF)  (COL(1),A);
    PUT EDIT ('OVHDEXPS     '  ^^ OVHDEXPS) (COL(1),A);
    PUT EDIT ('PBT          '  ^^ PBT)   (COL(1),A);
    PUT EDIT ('TAX          '  ^^ TAX)   (COL(1),A);
    PUT EDIT ('PAT          '  ^^ PAT)   (COL(1),A);
    PUT EDIT ('EARN         '  ^^ EARN)  (COL(1),A);
END; /* RN = 'MKTEST' */


ELSE IF RN = 'MKTHIST' THEN DO;
     DISPLAY ('PERIOD      ' ^^ PERIOD);
     DISPLAY ('YEAR1       ' ^^ YEAR1);
     DISPLAY ('YEAR2       ' ^^ YEAR2);
     DISPLAY ('YEAR3       ' ^^ YEAR3);
     DISPLAY ('YEAR4       ' ^^ YEAR4);
     DISPLAY ('YEAR5       ' ^^ YEAR5);
     DISPLAY (' ');
     PUT EDIT ('PERIOD      ' ^^ PERIOD) (COL(1),A);
     PUT EDIT ('YEAR1       ' ^^ YEAR1) (COL(1),A);
     PUT EDIT ('YEAR2       ' ^^ YEAR2) (COL(1),A);
     PUT EDIT ('YEAR3       ' ^^ YEAR3) (COL(1),A);
     PUT EDIT ('YEAR4       ' ^^ YEAR4) (COL(1),A);
     PUT EDIT ('YEAR5       ' ^^ YEAR5) (COL(1),A);
END; /* RN = 'MKTHIST' */
PUT SKIP;
END DBGET;


DBCHANGE: PROCEDURE;
DECLARE TCARD CHAR(72) VARYING EXTERNAL,
        TSTRING CHAR(12) VARYING EXTERNAL,
        DN CHAR(12) VARYING EXTERNAL,
        DVAL CHAR(24) VARYING EXTERNAL;
DECLARE T_CHG CHAR(36) INIT (' '),
        I FIXED DECIMAL;
DECLARE CHG_OK BIT(1) EXTERNAL,
        ANS CHAR(3);
 CHG_OK = TRUE;
DISPLAY
('DO YOU WANT TO CHANGE A VALUE IN ' ^^ RN ^^ ' RECORD?');
DISPLAY('YES/NO')
   REPLY (ANS);
DISPLAY(' ');
IF (ANS \= 'NO') THEN IF (ANS \= 'YES') THEN
```

```
    DISPLAY ('YOUR ONLY VALID ANSWERS ARE YES AND NO.')
    REPLY (ANS);
 DISPLAY (' ');
   IF ANS = 'NO' THEN CHG_OK = FALSE;
 DO WHILE (ANS = 'YES');
      DISPLAY ('TYPE THE NAME AND THE NEW VALUE FOR THE
ITEM');
      DISPLAY ('YOU WANT TO CHANGE, THEN PRESS ENTER')
        REPLY (T_CHG);
      DISPLAY (' ');
      TCARD = T_CHG;
      CALL GIT;
      DN = TSTRING;
      CALL GIT;
      DVAL = TSTRING;
      IF RN = 'MKTHIST' THEN IF (DN \= 'PERIOD') THEN
           DO I = 1 TO 3;
              CALL GIT;
              DVAL = DVAL ^^ ' ' ^^ TSTRING;
           END;
      CALL DBCHG_VAL;
      DN = ' '; DVAL = ' ';
      DISPLAY ('DO YOU WANT TO CHANGE ANY OTHER DATA
ITEMS');
      DISPLAY ('IN THE ' ^^ RN ^^ 'RECORD? YES/NO')
         REPLY (ANS);
      DISPLAY (' ');
      IF (ANS \= 'NO') THEN IF (ANS \= 'YES') THEN
         DISPLAY ('YOUR ONLY VALID ANSWERS ARE YES AND NO.')
         REPLY (ANS);
      DISPLAY (' ');
 END; /* DO WHILE ANS = 'YES' */
 IF (CHG_OK = TRUE) THEN CALL DBCHG_REC;
 END DBCHANGE;
   /************************************************/
 DBREPORT: PROCEDURE;
 DECLARE (ELAST,HLAST) BIT(1) INIT('0'B),
         MORE BIT(1) INIT('1'B);
 PUT EDIT
 ('PRODUCT      MARKET      MKTEST      MKHIST') (COL(1),A);
 PUT EDIT
 ('=======      ======      ======      ======') (COL(1),A);
 OBTAIN FIRST RECORD(PRODUCT) AREA (PRODUCT_REGION);
 CALL IDMS_STATUS;
 DO WHILE (ERROR_STATUS = OK);
      PRODBUF = PRODUCT_REC;
      OBTAIN FIRST RECORD (MARKET) SET (PROD_MKT);
      IF (ERROR_STATUS \= OK) THEN CALL DBPRINT
         (PRODBUF,MKTBUF,MKTESTBUF,MKTHISTBUF);
      DO WHILE (ERROR_STATUS = OK);
           MKTBUF = MARKET_REC;
           OBTAIN FIRST RECORD (MKTEST) SET (EST);
           IF (ERROR_STATUS \= OK) THEN ELAST = TRUE;
           ELSE MKTESTBUF = MKTEST_REC;
           OBTAIN FIRST RECORD (MKTHIST) SET (HIST);
           IF (ERROR_STATUS \= OK) THEN HLAST = TRUE;
           ELSE MKTHISTBUF = MKTHIST_REC;
           IF (ELAST = TRUE) THEN IF (HLAST = TRUE) THEN DO;
                CALL DBPRINT
(PRODBUF,MKTBUF,MKTESTBUF,MKTHISTBUF);
                MORE = FALSE;
           END;
           DO WHILE (MORE = TRUE);
                CALL DBPRINT
(PRODBUF,MKTBUF,MKTESTBUF,MKTHISTBUF);
                OBTAIN NEXT RECORD (MKTEST) SET (EST);
                IF (ERROR_STATUS \= OK) THEN ELAST = TRUE;
                ELSE MKTESTBUF = MKTEST_REC;
                OBTAIN NEXT RECORD (MKTHIST) SET (HIST);
                IF (ERROR_STATUS \= OK) THEN HLAST = TRUE;
                ELSE MKTHISTBUF = MKTHIST_REC;
                IF (ELAST = TRUE) THEN IF (HLAST = TRUE)
THEN
                   MORE = FALSE;
           END; /* DO WHILE ^(ELAST) OR ^(HLAST) */
           OBTAIN NEXT RECORD (MARKET) SET (PROD_MKT);
   ELAST = FALSE;
   HLAST = FALSE;
   MORE = TRUE;
      END; /* DO WHILE STILL MORE MARKET RECORDS IN THE SET
*/
      OBTAIN NEXT RECORD (PRODUCT) AREA (PRODUCT_REGION);
 END; /* DO WHILE STILL MORE PRODUCT RECORDS IN
PRODUCT_REGION*/
 END DBREPORT;
   /************************************************/
 STATUS_CHECK:  PROCEDURE(RN,RID);
 DECLARE RN CHAR(8) VARYING,
         RID CHAR(12) VARYING;
 DECLARE FOUND BIT(1) EXTERNAL;
 IF (ERROR_STATUS \= OK) THEN DO;
      DISPLAY ('THE ' ^^ RID ^^ ' ' ^^ RN ^^ 'RECORD');
      DISPLAY ('WAS NOT FOUND.');
      FOUND = FALSE;
 END;
 ELSE FOUND = TRUE;
 END STATUS_CHECK;
   /************************************************/
 REC_FIND:  PROCEDURE;
 DECLARE RN CHAR(8) VARYING EXTERNAL,
         RID CHAR(12) VARYING EXTERNAL,
         PID CHAR(12) VARYING EXTERNAL,
         MID CHAR(12) VARYING EXTERNAL;
 DECLARE FOUND BIT(1) EXTERNAL;
     FOUND = TRUE;
 IF RID = 'NEW_PRODUCT' THEN DO;
        OBTAIN FIRST RECORD(PRODUCT) AREA(PRODUCT-REGION);
        RN = 'NEW_PROD';
```

```
       CALL STATUS_CHECK(RN,RID);
END;
ELSE DO;
IF RN = 'PRODUCT' THEN DO;
     PRODNAME = RID;
     OBTAIN CALC RECORD (PRODUCT);
     CALL STATUS_CHECK (RN,RID);
     IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
END;
ELSE IF RN = 'MARKET' THEN DO;
   IF CTYP = 'DBADD' THEN DO;
     PRODNAME = PID;
     OBTAIN CALC RECORD(PRODUCT);
     CALL STATUS_CHECK(RN,RID);
     IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
   END;
 ELSE DO;
   MKTNAME = RID;
   OBTAIN CALC RECORD (MARKET);
     CALL STATUS_CHECK(RN,RID);
     IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
    OBTAIN OWNER SET(PROD_MKT);
    IF PRODNAME  \= PID THEN DO;
                            FOUND = FALSE;
                             GOTO WRONG;
                              END;
    ELSE OBTAIN CALC RECORD(MARKET);
END;
END;
ELSE IF RN = 'MKTEST' THEN DO;
    MKTNAME = MID;
    OBTAIN CALC RECORD(MARKET);
     CALL STATUS_CHECK(RN,RID);
     IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
    OBTAIN OWNER SET(PROD_MKT);
     IF PRODNAME \= PID THEN DO;
                             FOUND = FALSE;
                             GOTO WRONG;
                             END;
    ELSE OBTAIN CALC RECORD(MARKET);
   IF CTYP = 'DBADD' THEN;
   ELSE DO;
     OBTAIN FIRST RECORD (MKTEST) SET (EST);
     CALL STATUS_CHECK (RN,RID);
     IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
     DO WHILE (FDATE \= RID);
          OBTAIN NEXT RECORD (MKTEST) SET (EST);
          CALL STATUS_CHECK (RN,RID);
          IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
     END; /* * DO WHILE FDATE \= RID */
 END;
END; /* IF RN = 'MKTEST' */
ELSE IF RN = 'MKTHIST' THEN DO;
     MKTNAME = MID;
```

```
      OBTAIN CALC RECORD(MARKET);
      CALL STATUS_CHECK(RN,RID);
      IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
     OBTAIN OWNER SET(PROD_MKT);
     IF PRODNAME \= PID THEN DO;
                               FOUND = FALSE;
                                 GOTO WRONG;
                                  END;
     ELSE OBTAIN CALC RECORD(MARKET);
      OBTAIN FIRST RECORD (MKTHIST) SET (HIST);
     IF CTYP = 'DBADD' THEN;
     ELSE DO;
      CALL STATUS_CHECK (RN,RID);
      IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
       DO WHILE (PERIOD \= RID);
            OBTAIN NEXT RECORD (MKTHIST) SET (HIST);
            CALL STATUS_CHECK (RN,RID);
            IF FOUND = FALSE THEN GOTO REC_NOT_FOUND;
       END; /* DO WHILE PERIOD \= RID */
  END;
 END; /* RN = 'MKTHIST' */
 ELSE DO; DISPLAY  (RN ^^ 'IS NOT A CORRECT RECORD TYPE');
                          FOUND = FALSE;
                           END;
  WRONG:  IF FOUND = FALSE THEN DISPLAY ('RECORD NOT
FOUND');
 END;
 REC_NOT_FOUND:   ;
 END REC_FIND;
  /************************************************/
 GIT:  PROCEDURE;
 DECLARE TCARD CHAR(72) VARYING EXTERNAL,
        TSTRING CHAR(12) VARYING EXTERNAL;
 DECLARE (S,BL) FIXED DECIMAL (2,0);
 S = 1;  BL = 0;
  DO_CMD = TRUE;
 IF (LENGTH(TCARD) >= 5 ) THEN DO;
      DO WHILE (SUBSTR(TCARD,1,1) = ' ');
          TCARD = SUBSTR(TCARD,2);
          IF (LENGTH(TCARD) < 5) THEN DO;
                NOGIT = TRUE;
                GOTO CMD_NOT_FOUND;
          END;
      END; /* WHILE SUBSTR(TCARD,1,1) = ' ' */
      BL = INDEX(TCARD,' ');
      IF (BL = 0) THEN DO;
          TSTRING = TCARD;
          NOGIT = FALSE;
      END;
      ELSE DO;
          TSTRING = SUBSTR(TCARD,1,BL-1);
          TCARD = SUBSTR(TCARD,BL+1);
          NOGIT = FALSE;
      END;
 END;
 ELSE DO;
      DISPLAY (TCARD);
      DISPLAY ('CAN NOT BE PARSED.');
     DO_CMD = FALSE;
 END;
 CMD_NOT_FOUND: ;
 END GIT;

/******************************************************/
 DBPARSE_CMD: PROCEDURE;
 DECLARE TCARD CHAR(72) VARYING EXTERNAL;
 DECLARE TSTRING CHAR(72) VARYING EXTERNAL;
  DO_CMD = TRUE;
 CALL GIT;
 IF (NOGIT = TRUE) THEN DO; DISPLAY ('CTYP NOT FOUND');
                            GO TO INCOMPLETE;
                            END;
  ELSE CTYP = TSTRING;
  IF CTYP = 'DBADD' THEN DO;
           CALL GIT;
           CALL GIT;
           IF (NOGIT = TRUE) THEN DO;  DISPLAY ('RN NOT
FOUND');
                                       GO TO INCOMPLETE;
                                       END;
               IF TSTRING = 'NEW_PRODUCT' THEN DO;
                            RID = 'NEW_PRODUCT';
                            END;
               ELSE DO;
              RN = TSTRING;
              CALL GIT;
              CALL GIT;
              IF (NOGIT = TRUE) THEN DO;
                         DISPLAY ('PID NOT FOUND');
                         GO TO INCOMPLETE;
                         END;
                         ELSE DO;
                         RID = TSTRING;
                         PID = TSTRING;
                         END;
             IF RN = 'MARKET' THEN;
             ELSE DO;
             CALL GIT;
             CALL GIT;
             CALL GIT;
             IF (NOGIT = TRUE) THEN DO;
                       DISPLAY ('MID NOT FOUND');
                       GO TO INCOMPLETE;
                       END;
             ELSE MID = TSTRING;
        END;
    END;
 END;
 ELSE DO;
 IF CTYP = 'DBMENU' THEN;
 ELSE IF CTYP = 'DBREPORT' THEN;
 ELSE IF CTYP = 'LEAVE' THEN;
 ELSE DO;
      CALL GIT;
      CALL GIT;
      IF (NOGIT = TRUE) THEN DO; DISPLAY ('RID NOT FOUND');
                                GO TO INCOMPLETE;
                                END;
      ELSE RID = TSTRING;
      CALL GIT;
      IF (NOGIT = TRUE) THEN DO; DISPLAY ('RN NOT FOUND');
                                 GO TO INCOMPLETE;
                                 END;
      ELSE RN = TSTRING;
      IF RN = 'PRODUCT' THEN;
      ELSE IF RN = 'MARKET' THEN DO;
          CALL GIT;
          CALL GIT;
          IF (NOGIT=TRUE) THEN DO; DISPLAY ('PID NOT FOUND
');
                                   GO TO INCOMPLETE;
                                   END;
          ELSE PID = TSTRING;
      END;
      ELSE IF RN = 'MKTEST' THEN DO;
          CALL GIT;
          CALL GIT;
          IF (NOGIT = TRUE) THEN DO; DISPLAY ('PID NOT
FOUND');
                                     GO TO INCOMPLETE;
                                     END;
          ELSE PID = TSTRING;
          CALL GIT;
          CALL GIT;
          CALL GIT;
          IF (NOGIT = TRUE) THEN DO; DISPLAY ('MID NOT
FOUND');
                                    GO TO INCOMPLETE;
                                    END;
          ELSE MID = TSTRING;
      END;
      ELSE IF RN = 'MKTHIST' THEN DO;
          CALL GIT;
          CALL GIT;
          IF (NOGIT = TRUE) THEN DO; DISPLAY ('PID NOT
FOUND');
                                    GO TO INCOMPLETE;
                                    END;
          ELSE PID = TSTRING;
          CALL GIT;
          CALL GIT;
          CALL GIT;
          IF (NOGIT = TRUE) THEN DO; DISPLAY ('MID NOT
FOUND');
                                        GO TO INCOMPLETE;
                                        END;
          ELSE MID = TSTRING;
      END;
 END;
 END;
 INCOMPLETE:
  IF NOGIT = TRUE THEN DO;
             DISPLAY (TCARD);
             DISPLAY ('INCOMPLETE COMMAND');
             DO_CMD = FALSE;
             END;
 END DBPARSE_CMD;
   /*****************************************************/
 DBADD_REC:  PROCEDURE;
 DECLARE RN CHAR(8) VARYING EXTERNAL;
 IF RN = 'PRODUCT' THEN DO;
      STORE RECORD(PRODUCT);
      CALL IDMS_STATUS;
 END; /* RN = 'PRODUCT' */
 ELSE IF RN = 'MARKET' THEN DO;
      STORE RECORD (MARKET);
      CALL IDMS_STATUS;
 END; /* RN = 'MARKET */
 ELSE IF RN = 'MKTEST' THEN DO;
      STORE RECORD (MKTEST);
      CALL IDMS_STATUS;
 END; /* RN = 'MKTEST' */
 ELSE IF RN = 'MKTHIST' THEN DO;
      STORE RECORD (MKTHIST);
      CALL IDMS_STATUS;
 END; /* RN = 'MKHIST' */
 END DBADD_REC;

/*****************************************************/
 DBADD:  PROCEDURE;
 DECLARE TCARD CHAR(72) VARYING EXTERNAL;
 DECLARE TSTRING CHAR(12) VARYING EXTERNAL,
         DN CHAR(12) VARYING EXTERNAL,
         DVAL CHAR(24) VARYING EXTERNAL;
 DECLARE T_CHG CHAR (36) INIT (' ');
 DECLARE I FIXED DECIMAL;
 DECLARE CHG_OK BIT(1) EXTERNAL,
         ANS CHAR(3);
  CHG_OK = TRUE;
 DISPLAY ('DO YOU WANT TO ADD A NEW ' ^^ RN ^^ ' RECORD?');
 DISPLAY ('YES/NO')
    REPLY (ANS);
 DISPLAY (' ');
 IF (ANS \= 'NO') THEN IF (ANS  \= 'YES') THEN
    DISPLAY ('YOUR ONLY VALID ANSWER IS YES OR NO.')
    REPLY (ANS);
 DISPLAY (' ');
   IF (ANS = 'NO') THEN CHG_OK = FALSE;
 DO WHILE (ANS = 'YES');
       DISPLAY ('TYPE THE NAME AND THE NEW VALUE FOR AN
ITEM');
       DISPLAY ('IN THE NEW RECORD, THEN PRESS ENTER')
           REPLY (T_CHG);
       DISPLAY (' ');
       TCARD = T_CHG;
       CALL GIT;
       DN = TSTRING;
       CALL GIT;
       DVAL = TSTRING;
       IF RN = 'MKTHIST' THEN IF (DN \= 'PERIOD') THEN
             DO I = 1 TO 3;
                CALL GIT;
                DVAL = DVAL ^^ ' ' ^^ TSTRING;
             END;
       CALL DBCHG_VAL;
       DN = ' '; DVAL = ' ';
       DISPLAY ('DO YOU WANT TO ADD ANY OTHER DATA ITEMS');
       DISPLAY ('IN THE ' ^^ RN ^^ 'RECORD? YES/NO')
          REPLY (ANS);
       DISPLAY (' ');
       IF (ANS \= 'NO') THEN IF (ANS \= 'YES') THEN
          DISPLAY ('YOUR ONLY VALID ANSWER IS YES OR NO.')
          REPLY (ANS);
       DISPLAY (' ');
 END; /* DO WHILE ANS = YES */
 IF (CHG_OK = TRUE) THEN CALL DBADD_REC;
 END DBADD;

/***********************************************************
/
 PROC_CMD:  PROCEDURE;
 DECLARE CMD CHAR(72) EXTERNAL;
 DECLARE TCARD CHAR(72) VARYING EXTERNAL;
 DECLARE FOUND BIT(1) EXTERNAL;
   DECLARE CHG_OK BIT(1) EXTERNAL;
 TCARD = CMD;
 CALL DBPARSE_CMD;
 IF DO_CMD = TRUE THEN DO;
 IF CTYP = 'DBMENU' THEN CALL DBMENU;
 ELSE IF CTYP = 'DBREPORT' THEN DO;
      CALL DBREPORT;
      DISPLAY ('THE REPORT HAS BEEN SENT TO THE OUTPUT
FILE');
 END;
 ELSE IF CTYP = 'LEAVE' THEN;
 ELSE DO;
    CALL REC_FIND;
    IF (FOUND = TRUE) THEN DO;
        IF CTYP = 'DBGET' THEN DO;
             CALL DBGET(RN);
```

```
            DISPLAY ('THE ' ^^ RID ^^ ' ' ^^ RN ^^ '
RECORD');
            DISPLAY ('HAS BEEN SENT TO THE OUTPUT FILE');
        END;
        ELSE IF CTYP = 'DBDELETE' THEN DO;
            CALL DBDELETE (RN);
            DISPLAY ('THE' ^^ RID ^^ ' ' ^^ RN  ^^
'RECORD');
            DISPLAY ('HAS BEEN DELETED.');
        END;
        ELSE IF CTYP = 'DBCHANGE' THEN DO;
            CALL DBCHANGE;
            DISPLAY ('THE ' ^^ RID ^^ ' ' ^^ RN ^^ '
RECORD');
            DISPLAY ('HAS BEEN CHANGED');
        END;
        ELSE IF CTYP = 'DBADD' THEN DO;
        IF RID = 'NEW_PRODUCT' THEN DO;
                RN = 'PRODUCT';
                PRODIDNO = ' ';
                PRODNAME = ' ';
                PRODSTATUS = ' ';
       END;

       IF RN = 'MARKET' THEN DO;
              MKTIDNO = ' ';
              MKTNAME = ' ';
              OVHDCOST = ' ';
              OVHDRATE = ' ';
              TAXRATE  = ' ';
              SHARES   = ' ';
              UCOST    = ' ';
              PRICE    = ' ';
         END;
                  IF RN = 'MKTEST' THEN DO;
                     FDATE = ' ';
                     MKTSHR = ' ';
                     UNITS  = ' ';
                     SALES  = ' ';
                     COS    = ' ';
                     GPROF  = ' ';
                     OVHDEXPS = ' ';
                     PBT = ' ';
                     TAX = ' ';
                     PAT = ' ';
                     EARN = ' ';
                  END;
                  IF RN = 'MKTHIST' THEN DO;
                     PERIOD = ' ';
                     YEAR1  = ' ';
                     YEAR2  = ' ';
                     YEAR3  = ' ';
                     YEAR4  = ' ';
                     YEAR5  = ' ';
```

```
                    END;
            CALL DBADD;
            IF CHG_OK = TRUE THEN DO;
            DISPLAY ('A NEW RECORD OCCURRENCE HAS BEEN');
            DISPLAY ('INSERTED FOR THE ' ^^ RN ^^ '
RECORD');
            DISPLAY ('TYPE.');
            END;
      END;
      ELSE DO;
            DISPLAY (CTYP ^^ ' IS NOT A CORRECT');
            DISPLAY ('COMMAND TYPE.');
      END;
 END;
 END;
 CTYP = ' ';  RID = ' ';  RN = ' ';  PID = ' ';  MID = ' ';
 END;
 END PROC_CMD;
   /**************************************************/
 DBUILD:  PROCEDURE;
 /*CASE NO 'EOF' INCLUDED */
 ON ENDFILE (SYSIN) INREC = 'EOF';
 GET EDIT (INREC) (COL(1),A(80));
 DO WHILE (SUBSTR(INREC,1,3) \= 'EOF');
      RN = TYP;
      PUT EDIT (INREC) (COL(1),A);
      CALL DBINIT(RN);
      GET EDIT (INREC) (COL(1),A(80));
 END;
 END DBUILD;
   /**************************************************/
 DBMENU:  PROCEDURE;
 DECLARE NOTHING CHAR(8) INIT (' ');
 DISPLAY ('IF YOU ARE USING A COURIER TERMINAL PLEASE');
 DISPLAY ('PRESS PA 2 THEN ENTER.  OTHERWISE PRESS RETURN')
 REPLY (NOTHING);
 DISPLAY
 ('YOU ARE NOW RUNNING UNDER IDMS.  BELOW YOU WILL SEE
DISPLAYED A');
 DISPLAY
 ('MENU OF THE AVAILABLE DB MANIPULATION COMMANDS.  BEFORE
USING ANY');
 DISPLAY
 ('OF THE COMMANDS IN THE MENU IT IS RECOMMENDED THAN YOU
READ A ');
 DISPLAY
 ('DESCRIPTION OF EACH.  THIS INFORMATION IS AVAILABLE FROM
YOUR ');
 DISPLAY
 ('INSTRUCTOR.  IF YOU WANT TO LEAVE IDMS TYPE "LEAVE" THEN
PRESS');
 DISPLAY
 ('ENTER FOLLOWING ANY PROMPT FOR A COMMAND.');
 DISPLAY (' ');
```

```
DISPLAY
('     DBGET THE (RID) (RN) FOR (PID) IN THE (MID)');
DISPLAY (' ');
DISPLAY
('     DBDELETE THE (RID) (RN) FOR (PID) IN THE MID)');
DISPLAY
(' ');
DISPLAY
('     DBCHANGE THE (RID) (RN) FOR (PID) IN THE (MID)');
DISPLAY (' ');
DISPLAY
('     DBADD A (RN) FOR (PID) IN THE (MID)');
DISPLAY (' ');
DISPLAY (' ');
DISPLAY
('     DBMENU;     DBREPORT;     LEAVE;     ');
DISPLAY (' ');
DISPLAY
('TO EXECUTE A COMMAND TYPE THE COMMAND WITH THE ');
DISPLAY
('APPROPRIATE ARGUMENTS THEN PRESS RETURN.');
END DBMENU;

END DBQUERY; /* END MAIN PROGRAM */
```

TABLE 3

Storage requirements for IDMS (Data Dictionary in Packed Format)

Filename	Filetype	LRCL	RECORDS	BLOCKS
LOAD	MAP	100	131	17
CHEMDEV1	SCHMA	80	194	20
DATABASE	FILE1	496	30	19
DBOUT	LISTING	81	213	14
SESSION	GLOBALV	24	257	7
DLODDB	DB	800	21	21
DMSGDB	DB	800	12	12
DICTDB	DB	800	343	343
FILE	SYSLST	133	1	1
DEBQUERY	DMLP	80	949	95
DBIN2	DECK	80	34	4
DEBQUERY	LISTING	800	104	104
DEBQUERY	TEXT	80	743	75
DBENTER	EXEC	65	16	1
DEVDMCL	LISTING	133	35	6
NUPROD	LISTING	133	48	8
NUPROD	TEXT	80	44	5
DEVDMCL	DMCL	80	28	3
DEVDMCL	TEXT	80	30	3
NUPROD	SUBSC	80	17	2
Totals			3250	754

TABLE 4

Storage Requirements for IDMS files (Data Dictionaries unpacked)

Filename	Filetype		LREC	RECORDS	BLOCKS
DMSGDB	DB	A1 F	2496	200	624
DICTDB	DB	A1 F	3664	1000	4580
DLODDB	DB	A1 F	2496	400	1248
LOAD	MAP	A5 F	100	131	17
CHEMDEV1	SCHMA	A1 F	80	194	20
DATABASE	FILE1	A1 F	496	30	19
DBOUT	LISTING	A1 V	81	213	14
DBINIT	LISTING	A1 F	133	17	3
SESSION	GLOBALV	A1 V	24	257	7
FILE	SYSLST	A1 F	133	1	1
DEBQUERY	DMLP	A1 F	80	949	95
DBIN2	DECK	A1 F	80	34	4
DEBQUERY	LISTING	A1 F	800	104	104
DEBQUERY	TEXT	A1 F	80	743	75
DBENTER	EXEC	A1 V	65	16	1
DEVDMCL	LISTING	A1 F	133	35	6
NUPROD	LISTING	A1 F	133	48	8
NUPROD	TEXT	A1 F	80	44	5
DEVDMCL	TEXT	A1 F	80	30	3
DEVDMCL	DMCL	A1 F	80	28	3
NUPROD	SUBSC	A1 F	80	17	2
Totals				4491	6909

TABLE 5

Storage Requirements for EMPIRE SYSTEM Files

Filename	Filetype	LRECL	RECORDS	BLOCKS
MODEL2	EMPMOD	80	42	5
MODEL5	TEXT	80	58	6
EMP$IRE	EXEC	75	57	2
MODEL5	FORTRAN	80	77	8
LOAD	MAP	100	346	44
CONTROL5	EMPCON	80	21	3
DATA5	EMPDATA	80	5	1
MODEL2	TEXT	80	58	6
CONTROL2	EMPCON	80	22	3
REPORT2	EMPREP	80	18	2
FSUM	EMPCON	80	16	2
REPORT5	EMPREP	80	18	2
MODEL5	EMPMOD	80	42	5
FINANSUM	TEXT	80	56	6
MODEL1	FORTRAN	80	54	6
MODEL2	FORTRAN	80	77	8
REPORT1	EMPREP	80	15	2
MODEL1	EMPMOD	80	33	4
DATA2	EMPDATA	80	5	1
FINANSUM	FORTRAN	80	53	6
REP	EMPREP	80	34	4
FSUMDATA	EMPDATA	80	14	2
FINANSUM	EMPMOD	80	50	[illegible]
totals			1171	134

APPENDIX D

MANGT 466
MANAGEMENT INFORMATION SYSTEMS
INSTRUCTOR: CONSTANZA CASTRO
SPRING/84

CATALOG DESCRIPTION: A comprehensive view of the organization's information requirements and the role of the computer information systems in gathering and producing information. Concepts of data resource management, assessing developments in information technology, and information system's impact on organizations. Problems and techniques concerning the development and installation of responsive systems with special attention to manager's use of system's outputs. Case studies and selected applications.
PR: CMPSC 202, FINAN 450, MANGT 420 and MKT6 400.

COURSE APPROACH

We will use the case study method to provide the opportunity to apply the systems approach in solving MIS problems. In addition, you will be given hands-on experience in using computer-based decision support system (DSS) models. We will rely on the textbook and.the lecture to convey the body of knowledge, but the important learning experience will be your direct involvement in problem solving. This problem-solving emphasis will be incorporated into our classroom sessions as we discuss case problems.

This course is treated as a capstone course in business computing science, tying together all of the material you have previously learned, and applying it in a problem-solving situation.

If you have not satisfied the prerequisites do not attempt to take this course thinking that you can pick up that material as we go along. The workload will not permit such an approach. We will proceed at a pace assuming that everyone has a good foundation knowledge of hardware, software, and systems.

TEXTBOOK

Management Information Systems, Second Edition, McLeod, Science Research Associates, 1983.

COURSE GRADE

Your course grade will be based on the following components and weights:

COMPONENT	POINTS	WEIGHT
Midterm exam	100	25%
Written case and DSS model Assignments	200	50%
Final Exam	100	25%
	400	100%

Both the midterm and final will be objective questions (true-false and multiple choice) on textbook material.

OFFICE HOURS

My office is in room 117D Calvin Hall. My phone is 532-5559.
My office hours are: 9:00-11:00 T-T

If these hours are not convenient with you, we can make other arrangements.

The key to making a good grade is knowing when you need help and asking for it. Let me know if you start to have difficulty.

CASE ASSIGNMENTS AND COMPUTER WORK:

The cases and computer assignments will be given enough time in advance so that you have an opportunity to do well on them. The computer assignemnts won't require knowledge of a specific computer language, but knowledge of the logic of programming will be helpful.

COURSE OUTLINE

Week	Topic/Activity
1	Course Introduction Chapter 1--Introduction to Information Management
2	Chapter 2--Theory of Management and Organization Chapter 3--The General Systems Model
3	Chapter 4--The Systems Approach Case Book Chapter 1--Solution of a Case Problem
4	Review of Computer hardware--selected portions of Chapters 5-7, 9, 10
5	Chapter 8--The Date Base
6	Chapter 11--Computer-Based Decision Support Systems
7	Midterm exam (weeks 1-6 material) Chapter 12--Introduction to Functional Information Systems
8	Chapter 13--Marketing Information Systems
9	Chapter 14--Manufacturing Information Systems
10	Chapter 15--Financial Information Systems
11	Chapter 16--The Planning Phase
12	Chapter 17--The Analysis and Design Phase
13	Chapter 17 (Continued) Chapter 18--The Implementation Phase
14	Chapter 19--Operation and Control of the MIS
15	Chapter 20--The Future of the MIS
	Final Exam (weeks 7-15 material)

STUDENT BEHAVIORAL LEARNING OBJECTIVES (SBLO) SPECIFIED BY NEIL STRUNK IN HIS 1982 MASTER'S REPORT "SUPPORT TOOLS FOR AN UNDERGRADUATE MANAGEMENT INFORMATION SYSTEM COURSE" [4]

Four General areas in Which MIS students should display certain behavioral responses are:
1. data base management systems,
2. office automation systems,
3. decision support systems,
4. transaction processors.

In the area of data base management systems, Business students should achieve certain abilities using one data base and one data base management system. On the predefined data base, the Business students should:

- use a query language to retrieve a signal record,
- use a query language to retrieve information from two types of records,
- add a record to a data base,
- delete a record from a data base,
- change a value in a record in a data base.

Information Systems students with no data base background should achieve these same behavioral objectives. With data base experience, the students should also change the structure in some predefined area of a data base.

Office automation systems is an area where Business students should interact with a computerized office system and demonstrate how to:

- create a text file,
- edit that text file,
- print a hard copy of that file,
- send an receive a message on a mail system,
- retrieve a file from an automatic file system,
- put a weeks schedule on a calendar system,
- schedule a meeting with a specifed number of people using the calendar system,

- set up and participate in an electronic conference,

- make a remote presentation.

Information Systems students with and without data base experience should demonstrate all of the above behavioral objectives except creating, editing, and printing of a text file. These text manipulation activities would be mastered in their other interactive programming activities. Information Systems students should also write a small menu driven subprogram for an available transaction processor.

Decision support systems is an important area where the ability to demonstrate the following objectives will give the Business student an appreciation for the power of a decision support system to integrate an organization's information. The Business students should achieve the following objectives:

- structured problem -- use a predefined application program to get the answer to a problem,

- semi-structured problem -- use one or more preprogrammed decision support tools, like forecasting programs, and combine the resulting incomplete pieces of information to formulate a decision to a predefined problem.

Information systems students without data base experience should achieve these same behavioral objectives. If the students have data base experience then they should do the following:

- structured problem -- use a predefined application program to get the answer to a problem,

- semi-structured problem -- build a semi-structured problem within a predefined scope,

- unstructured problem -- actively retrieve several pieces of information from different records in a data base, and use this information to formulate an answer to an unstructured problem.

Transaction processors are in wide use at the operational level of many organizations. Business and Information Systems students should be able to demonstrate the use of some preprogrammed transaction processor.

Table 6:

Summary of Implemented SBLO

Decision Support Systems

Decision Support Systems (DSS) are interactive computer systems. They are designed as an aid to managers in solving problems that involve judgment (i.e., unstructured and structured problems).

The SBLO for DSS are as follows:

- use an available application program to answer a structured problem.
- use one or more available DSS tools and combine the pieces to make a decision about a predefined problem.
- design a semi-structured problem environment within a predefined scope [5].

Database Management Systems

Objectives defined in the area of Database Management Systems will make use of one database and one database management system. The database will already be available and the SBLO are:

- use a query language to retrieve records.
- use a query language to retrieve information from different types of records
- add records to the database.
- delete records from a database.
- change field values in records in the database. [5].

The database will be provided to the students using IDMS (Integrated Data Management System, Cullinet Corporation) available through the National 6/30. Correct implementation of the above mentioned SBLO constitutes completion of the task.

Furthermore, the students will be required to explore a third area, namely:

Text Editing

The SBLO for this particular task will involve:

- using SCRIPT (text editing tool available through the National 6/30) to edit and format a predefined document.
- Route the formatted document through the system.
- Produce output at one of the printing stations according to specifications.

Completion of these steps will accomplish the SBLO for the task.

Case Analyses

Three case analyses will be required during the semester. Specific instructions as to the format and content of the finished analysis will be given. The students will work in teams to do the actual analysis work. These teams will be composed of a maximum of three students each, to provide the best chance for participation in discussion and cooperation in the production of the final document. There will be in-class discussion of the several solutions presented by the different groups.

The cases cover 3 main areas:

1. Problems found in updating of a data processing operation into an MIS.
2. Dealing with problems caused by lack of proper planning and implementation of a computerized MIS.

3. Using systems flow documentation to represent an existing system.

Lectures

The topics will give comprehensive coverage of the following areas:

1. MIS and DSS components.
2. General systems and management concepts.
3. Computerized systems components.
4. Planning and Implementation of an MIS.
5. Use of the MIS at different levels of management.

EMPIRE

The EMPIRE assignments helped you to:

1. Understand how Decision Support Systems (DSS) work.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

2. Analyze problem data to produce valuable information.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

3. Understand how a decision may be improved by using DSS.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

4. Increase your confidence in working with interactive DSS.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

5. Learn how to interpret output from DSS tools.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

IDMS

The IDMS assignment helped you to:

1. Learn how a query language works

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

EVALUATION FORM FOR MIS CLASS (MANAGEMENT 466)
SPRING SEMESTER, 1984

SOFTWARE ASSIGNMENT EVALUATION

SCRIPT

The SCRIPT assignment helped you:

1. Learn how to sign on to the mainframe computer at KSU using CMS.

________	________	________	________	________
Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

2. Learn how to use the XEDIT editing environment.

________	________	________	________	________
Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

3. Gain confidence with interactive systems.

________	________	________	________	________
Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

4. Learn how to obtain typed output at remote laboratory.

________	________	________	________	________
Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

5. Learn SCRIPT commands to produce desired format for output

________	________	________	________	________
Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

2. Learn to interact with a Database Management System (DBMS).

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

3. Learn how to integrate DBMS and DSS tools (i.e., see how the database is updated to reflect the latest decisions made by management.)

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

4. Learn how to add, delete, change, and retrieve records in a database.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

5. Learn how to produce reports reflecting your DBMS transactions.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

LECTURES

Lectures helped you to understand:

1. MIS and DSS elements, concepts, and functions.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

2. General Systems and management concepts.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

3. Computerized systems concepts.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

4. Planning, analysis and design, implementation, and operation of an MIS.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

5. Use of an MIS at different levels of management.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

CASES

The cases helped you visualize the potential problems found in:

1. Updating of a data processing operation into an MIS.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

2. Situations where there are indications of lack of proper planning and implementation of a computerized MIS.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

3. Using systems analysis documentation tools to represent an existing system.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

ASSIGNMENT DOCUMENTATION

The documentation and instructions given for the assignments:

1. Made assignments easy to fulfill.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

2. Were thorough and descriptive.

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree

SUGGESTIONS

What would be your suggestions for improvements of class content in future semesters?

BIBLIOGRAPHY

1. Kansas State University Bulletin, General Catalog 1983-84, pp. 204-205.

2. Statistical Abstract of the United States 1980 (Washington, DC.: U.S. Department of Commerce, Bureau of the Census, 1980), p. 570.

3. McLeod Jr., Raymond. Management Information Systems, 2nd ed. Science Research Associates, 1983.

4. Strunk, Neal V. Support Tools for an Undergraduate Management Information Systems Course. MS report, KSU 1982.

5. Birchard, Robert A. Design and Implementation of Problem Environments and Software Support Tools for a Management Information Systems Course. MS report, KSU 1983.

6. McLeod Jr., Raymond. The Undergraduate MIS Course at A.A.C.S.B. Schools: A Special Report to Participating Deans and Professors. (Texas A & M University, January 1984).

7. Tull, Donald S., and Del I. Hawkins. Marketing Research, Macmillan Publishing Co., Inc., 1976.

8. Applied Data Research. EMPIRE Decision Support System, Beginner's Guide. January, 1980 Draft.

9. Applied Data Research, Inc. EMPIRE Modeling, Analysis, and Reporting system: An Introduction. September, 1978.

10. Applied Data Research, Inc. EMPIRE Decision Support System User Reference Manual. September, 1980.

11. Applied Data Research, Inc. EMPIRE Decision Support System VM/370 User Guide. P12A-VM-01.

12. Conrow, Kenneth. The CMS Cookbook. Manhattan, Kansas: Computing Center, Kansas State University. December, 1982.

13. Cullinet Corporation. IDMS Data Definition Languages, Utilities, and GCI Reference Guide. Release 3.1, revision 1. April, 1975.

14. Cullinet Corporation. IDMS Database Design and Definition Guide. Release 4.0, revision 4. February, 1977.

15. Cullinet Corporation. IDMS Programmers Reference Guide. Release 4.5. August, 1977.

16. International Business Machines. OS and DOS PL/I Language Reference Manual. 1st ed., September, 1981.

17. International Business Machines. OS PL/I Optimizing Compiler: Programmers Guide. Release 4.0, 1981.

18. International Business Machines. IBM virtual Machine/System Product: CMS Command and Macro Reference. 1st ed., September 1980.

19. International Business Machines. IBM Virtual Machine/System Product: System Programmer's Guide. 1st ed., September 1980.

IMPLEMENTATION OF PROBLEM ENVIRONMENTS AND SOFTWARE SUPPORT TOOLS FOR A MANAGEMENT INFORMATION SYSTEMS COURSE FOR STUDENTS IN THE COLLEGE OF BUSINESS ADMINISTRATION AT KANSAS STATE UNIVERSITY

by

CONSTANZA CASTRO

B.S., University of Oregon, 1975
MBA, Kansas State University, 1976

AN ABSTRACT OF A MASTER'S THESIS

Submitted in partial fulfillment of the

requirements for the degree of

MASTER of SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

This thesis describes the implementation of problem environments and software support tools for a Management Information Systems course to be offered for all business majors in the College of Business at Kansas State University.

Two areas were chosen from the Student Behavioral Learning Objectives (SBLO) developed in Neil Strunk's 1982 master's project entitled "Support Tools for an Undergraduate MIS course" to implement problem environments. These areas were Decision Support Systems (DSS) and Database Management Systems (DBMS). Another set of SBLO for text editing was developed, and lectures and case studies were designed for inclusion in the course.

The database and user "front end" developed by Robert A. Birchard in a 1983 master's report titled "Design and Implementation of Problem Environments and Software Support Tools for a Management Information Systems Course" were modified to perform as required by the course objectives. Assignments were designed to make it possible to fulfill the chosen SBLO effectively.

The EMPIRE system was used for the DSS problem environments and the IDMS DBMS for the DBMS problem environments. The SCRIPT system was used for text editing. All these software tools were implemented through the CMS environment on the Kansas State University mainframe computer (National 6/30).